



FH MÜNSTER
University of Applied Sciences



Deutsches Zentrum
für Luft- und Raumfahrt
German Aerospace Center

Konstruktion eines Phantoms zur Validierung der Quick Release- und Sinusoidal Perturbation-Messmethode am Muscle Atrophy Research and Exercise System

Ivo Gottschalk

Matrikelnummer: 970074

Milan Vincentz

Matrikelnummer: 776102

MASTERARBEIT IM STUDIENGANG BIOMEDIZINISCHE TECHNIK

Erstgutachter: Prof. Dr. Klaus Peikenkamp

Zweitgutachter: Prof. Dr. Jörn Rittweger

Köln, 21. Dezember 2020

Abstract

Ziel Quick Release- und Sinusoidal Perturbation-Messmethoden werden zur Bestimmung der mechanischen Eigenschaften der Muskeln verwendet. Dies ist vorallem im Hinblick auf die Muskelatrophie relevant, die während und nach längeren Aufenthalten im Weltraum auftritt. Dabei wird als Parameter häufig die muskulotendinöse und muskuloarikuläre Steifigkeit gemessen. Die beiden Messmethoden zur Bestimmung der Muskelsteifigkeit wurden jedoch nicht ausreichend validiert. Die Validierung der beiden Messmethoden erfolgt in dieser Arbeit durch die Konstruktion eines Phantoms.

Methodik Es wird ein Phantom mit bekannten Parametern konstruiert, welches die viskoelastischen Eigenschaften des M. triceps surae abbildet. Der M. triceps surae ist der primäre Plantarflexor des Fußes und klassisch stark von pathologischen Einflüssen der Mikrogravitation betroffen. An dem konstruierten Phantom können zur Validierung der Messmethoden Daten mit Hilfe von Quick Release und Sinusoidal Perturbation generiert und mit den theoretischen Daten des Phantoms verglichen werden.

Ergebnisse Die Ergebnisse der Quick Release Messmethode zeigen eine Abhängigkeit der Muskelsteifigkeit vom angelegten Drehmoment. Besonders bei sehr kleinen Drehmomenten wurden sehr hohe Steifigkeiten und damit Abweichungen vom Erwartungswert der Drehfederkonstante detektiert. Die Nutzung der Quick Release Methode lieferte bei Drehmomenten von 30-40 Nm Ergebnisse der höchsten Genauigkeit. Eine tiefergreifende Validierung der Testmethodik mit höheren Drehmomenten ist sinnvoll, die vorliegende Arbeit bietet jedoch eine adäquate Grundlage für zukünftige Untersuchungen.

Im Rahmen der Sinusoidal Perturbation-Testung konnte gezeigt werden, dass die Experimentaldaten nicht mit einem Modell zur Charakterisierung von Gelenkdynamiken übereinstimmt. Es wurden Abweichungen festgestellt, die auf externe Einflüsse hindeuten, weshalb eine tiefergreifende Untersuchung der Messmethodik sinnvoll erscheint. Die grundsätzliche Herangehensweise zur Validierung der Messmethodik ist jedoch vielversprechend und kann als robuste Grundlage für weitere Untersuchungen genutzt werden.

Durch die durchgeführten Tests zu QR und SP wurden zudem zahlreiche offene Fragen im Messsystem MARES aufgedeckt, die ein Verbesserungspotential für zukünftige Studien aufzeigen.

Inhaltsverzeichnis

Abstract	I
Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Abkürzungsverzeichnis	IX
1 Einleitung	1
1.1 Wissenschaftlicher Kontext - Herausforderungen der Raumfahrt	1
1.2 Zielsetzung und Relevanz - Validität der Messmethoden	2
1.3 Aufbau der Thesis	3
2 Theoretische Grundlagen	4
2.1 Stand der Forschung	4
2.1.1 Literatursuche	4
2.1.2 Einordnung in den wissenschaftlichen Kontext	5
2.2 Skelettmuskulatur	6
2.2.1 Grundlegende Anatomie des Skelettmuskels	7
2.2.2 Sehnen	7
2.2.3 Aponeurose	9
2.2.4 Musculus triceps surae	10
2.2.5 Mechanische Eigenschaften des Muskel-Sehnen-Komplexes	11
2.3 Artificial Gravity Bed Rest study with ESA	12
2.4 Muscle Atrophy Research and Exercise System	12
2.5 Methoden zur Charakterisierung von Gelenkdynamik	16
2.6 Quick Release	17
2.7 Sinusoidal Perturbation	20
2.8 Modell zur Darstellung viskoelastischen Verhaltens	24
3 Methodik	25
3.1 Quick Release	25
3.2 Sinusoidal Perturbation	26

3.3	Das SPQR-Phantom	26
3.3.1	Lastenheft des Phantoms	27
3.3.2	Designprozess und Entwürfe	28
3.3.3	Finales Design des SPQR-Phantoms	31
3.4	Parameterabschätzung und -validierung	34
3.4.1	Massenträgheitsmoment I	35
3.4.2	Rotatorische Dämpfungskonstante B	37
3.4.3	Drehfederkonstanten K	38
3.5	Testreihen	38
3.5.1	Quick Release	39
3.5.2	SP-Testreihen	40
3.6	Datenverarbeitung	41
3.6.1	Grundlagen der Auswertungsmethodik des Quick Release	42
3.6.2	Python-Automatisierung der Quick Release Auswertung	43
3.6.3	SP-Python-Automatisierung	47
4	Ergebnisse	51
4.1	Eingangsparameter	51
4.1.1	Massenträgheitsmoment	51
4.1.2	Dämpfungskonstante	52
4.1.3	Drehfederkonstanten	53
4.2	Quick Release	54
4.2.1	Bestimmung der Federsteifigkeit K	55
4.2.2	Einfluss des voreingestellten Trägheitsmoment im MARES	62
4.2.3	Muskelsteifigkeiten der AGBRESA-Studie	63
4.2.4	Federsteifigkeiten mit Auswertung durch Cosinus-Fit-Annäherung	65
4.3	Sinusoidal Perturbation	67
4.3.1	Typinterne Analyse	67
4.3.2	Vergleich des Phantoms mit dem mathematischen Modell	74
5	Diskussion	78
5.1	Validierung der grundlegenden Parameter	78
5.1.1	Überprüfung des Trägheitsmoments	78
5.1.2	Abweichungen der Dämpfungskonstanten	79
5.1.3	Abweichung der ermittelten Federkonstanten von den Herstellerangaben	79
5.2	SPQR-Phantom	79
5.2.1	Erfolgreiche Aspekte des Designs	80
5.2.2	Verbesserungspotential des Phantoms	81
5.3	MARES	82
5.4	Testreihen	84

5.5	Quick Release	85
5.5.1	Auswertungsmethodik des Quick Release	85
5.5.2	Bestimmung der Federsteifigkeit K	86
5.5.3	Einfluss des voreingestellten Trägheitsmoment im MARES	88
5.5.4	Federsteifigkeiten der AGBRESA Studie	90
5.5.5	Federsteifigkeiten mit Auswertung durch Cosinus-Fit-Annäherung	90
5.6	Sinusoidal Perturbation	91
5.6.1	Phantominterne Analyse	92
5.6.2	AGBRESA-interne Analyse	92
5.6.3	Überprüfung der Validität des mathematischen Modells	93
5.7	Vergleich der Steifigkeiten zwischen Modell, QR und SP	96
5.8	Entwurf eines alternativen Messsystems	98
6	Fazit und Ausblick	100
	Danksagung	102
	Eidesstattliche Erklärung	103
	Literatur	104
A	Anhang	112
A.1	Technische Spezifikationen des MARES	112
A.2	Steifigkeiten zu alten Endstoppositionen	114
A.3	Verwendeter Code	114
A.3.1	QR	114
A.3.2	SP	132

Abbildungsverzeichnis

2.1	Muskelaufbau im Querschnitt	7
2.2	Schematischer Aufbau der Sehne	8
2.3	Schematische Darstellung des <i>M. triceps surae</i>	11
2.4	MARES an Bord der ISS	13
2.5	Aufbau des Muscle Atrophy Research And Exercise System	16
2.6	Fußergometer zur Messung der Plantarflexion und -extension	18
2.7	Klassischer Winkelverlauf eines Quick Releases aufgetragen über der Zeit	20
2.8	Frequenzabhängige Gelenksteifigkeitsantwort	22
2.9	Beispielhafter Amplitudengang des muskuloartikulären Systems.	23
3.1	Skizzen der verworfenen Designs des SPQR-Phantoms	29
3.2	Leverkonfiguration des MARES mit Fixing-Adapttern für das SPQR-Phantom	31
3.3	SPQR-Phantom-Aufsätze für das MARES	32
3.4	SPQR-Aufsätze	33
3.5	Das MARES mit befestigtem SPQR-Phantom	34
3.6	Messaufbau zur Bestimmung des Massenträgheitsmoments des Phantoms.	37
3.7	Datenstreams zu Angular_Position und Subject_Torque	42
3.8	QR-Darstellung nach Anwendung des Python-Skriptes <i>mares_QRanalysis4 – 0.py</i>	44
3.9	QR-Darstellung nach Anwendung des Python-Skriptes <i>mares_QRanalysis3 – 7_delta.55 – cosfit.py</i>	47
4.1	Geradengleichungen zur Federkonstantenvalidierung	53
4.2	Relation von ermittelter Steifigkeit zu angelegtem Drehmoment	55
4.3	Prozentuale Abweichung der Federsteifigkeiten	57
4.4	Intervall-Standardabweichung der Federn	58
4.5	Variationskoeffizient aller Federn in Relation zum angelegten Drehmoment ohne Einfluss eines Dämpfers.	59
4.6	Relation aus Steifigkeit und Drehmoment	60
4.7	Abweichung der Federsteifigkeiten zu Drehfederkonstanten bei Dämpfer Aufbau	61
4.8	Variationskoeffizient aller Federn in Relation zum angelegten Drehmoment mit Einfluss eines Dämpfers.	61
4.9	Darstellung der ermittelten Trägheitsmomente für alte und neue voreingestellte Trägheit	62

4.10	Darstellung der Steifigkeit-Drehmoment-Relation für die ausgewählten AGBRESA Probanden A-H.	63
4.11	Vergleich der Steifigkeiten nach alter und neuer Fit-Methode der vier Federn . .	65
4.12	Vergleich der Steifigkeiten nach alter und neuer Fit-Methode bei AGBRESA . . .	66
4.13	Bode-Plot Phantom Muskelaktivierung %MVC	68
4.14	Bode-Plot Phantom Vorspannung MVC	69
4.15	Bode-Plot Phantom Drehfederkonstanten K	70
4.16	Bode-Plot Phantom Drehdämpfungskonstante B	71
4.17	Bode-Plot AGBRESA Muskelaktivierung %MVC	72
4.18	Bode-Plot AGBRESA Vorspannung MVC	73
4.19	Bode-Plot Phantom und math. Modell, $B = 0 \text{ Nmsrad}^{-1}$	74
4.20	Bode-Plot Phantom und math. Modell, $B = 105 \text{ Nmsrad}^{-1}$	75
4.21	Extrapolation nach Frequenz 0 Hz, $B = 0 \text{ Nms/rad}$	76
4.22	Extrapolation nach Frequenz 0 Hz, $B = 105 \text{ Nms/rad}$	76
5.1	Einfluss von Dämpfung auf den Kurvenverlauf eines Systems zweiter Ordnung . .	94
A.1	Steifigkeiten zur alten Endstopposition	114

Tabellenverzeichnis

3.1	Testreihen der QR-Untersuchungen	39
3.2	Exemplarische Darstellung des Messprotokolls für ausgewählte QR-Testläufe . . .	40
3.3	Auszug der Testreihen der SP-Untersuchungen	41
3.4	CSV-Ausgabe des Python-Skriptes <i>mares_QRanalysis4 - 0.py</i>	45
4.1	Ergebnisse der Federkonstantenvalidierung	53
4.2	Anzahl gemessener Quick Releases	54
4.3	Arithmetisches Mittel aller Federsteifigkeiten	56
4.4	Arithmetisches Mittel der Steifigkeitsabweichungen	58
4.5	Mittelwert mit Standardabweichungen der AGBRESA Probanden A-H mit jewei- liger Korrelation der Steifigkeiten zum angelegten Drehmoment.	64
4.6	Mittlere Steifigkeiten der vermessenen Federn	66
4.7	SP-Gain-Werte des Modells und des Phantoms bei Frequenz 0 Hz	77
5.1	Vergleich der federabhängig ermittelten Steifigkeitswerte aus SP bzw. QR	96

Abkürzungsverzeichnis

AGBRESA ..	Artificial Gravity Bed Rest study with ESA
ATR	Angle Torque Relation
BMU	Basic Motion Unit
CSV	Comma-seperated values
dB	Dezibel
DMS	Dehnungsmessstreifen
EMG	Elektromyografie
ESA	European Space Agency
HRF	Human Research Facility
Hz	Hertz
ISS	International Space Station
ISPR	International Standard Payload Rack
kg	Kilogramm
KV	Kelvin-Voigt
m	Meter
MARES	Muscle Atrophy Research and Exercise System
MVC	Maximum Voluntary Contraction
Nm	Newtonmeter
QR	Quick Release
rad	Radiant
ROM	Range Of Motion

s Sekunde

SEC Series Elastic Component

SLA short lever-Arm

SP Sinusoidal Perturbation

STS Space Transportation System

Kapitel 1

Einleitung

1.1 Wissenschaftlicher Kontext - Herausforderungen der Raumfahrt

Der erste Mensch im Weltraum war der russische Kosmonaut Yuri Alekseyevich Gagarin am 12. April 1961. Er verbrachte eine Stunde und 46 Minuten im Orbit der Erde bevor er sicher zur Erde zurückkehrte [1]. Seitdem haben sich nicht nur die technischen Aspekte der Raumfahrt weiterentwickelt, sondern auch die Erforschung der gesundheitlichen Effekte und schädlichen Einflüsse auf die Gesundheit der Astronauten. Auch im Hinblick auf zukünftige Missionen zu Mond und Mars, bei denen Astronauten potenziell mehrere Jahre der Mikrogravitation des Weltraums ausgesetzt sind, ergeben sich neue Herausforderungen, die mit aktueller Technik noch nicht zu bewältigen sind.

Das Institut für Luft- und Raumfahrtmedizin des Zentrums für deutsche Luft- und Raumfahrt befasst sich daher mit den Herausforderungen, die Astronauten bewältigen müssen, wenn sie auf Langzeitmissionen einer schwerelosen Umgebung ausgesetzt sind.

Die Effekte, die die Raumfahrt auf die Körper der Astronauten hat, wurden von Hodkinson et al. zusammengefasst [2].

Die Mikrogravitation während längerer Weltraumaufenthalte führt dabei zu Veränderungen in beinahe allen physiologischen Systemen des Körpers [3]. Einer der Haupteffekte der Mikrogravitation ist eine durch die fehlende mechanische Belastung der Muskeln und Knochen bedingte Atrophie der Muskeln und einem einhergehenden Verlust an Muskelkraft, Muskelfunktion und Motorik [4][5], vor allem in den unteren Extremitäten. Weitere Effekte sind neben Gangstörungen [6][7], zerebralen- [8] und kardiovaskulären Veränderungen [9][10][11] auch eine Verschiebung des Flüssigkeitsvolumens in Richtung des Thorax, was Einfluss auf verschiedene Körperfunktionen nimmt [12].

Die Mikrogravitation hat zudem einen Einfluss auf die mechanischen, spezieller die kontraktile und viskoelastischen Eigenschaften der Muskeln. Diese werden durch die Muskelsteifigkeit charakterisiert, welche weit weniger im Fokus wissenschaftlicher Untersuchungen steht, als andere Muskelparameter. Dabei ist die Steifigkeit ein wichtiger Parameter für die Kontrolle von Bewegungen, da sie einen Widerstand gegen externe Störungsfaktoren darstellt und eine erhöhte Steifigkeit das Ausführen von Bewegungen erschweren kann [13]. Dadurch könnten Veränderungen der Steifigkeit sich im Arbeitsalltag einer Raumstation als kritisch erweisen.

Mit der Entdeckung von mehr und mehr negativen Effekten der Mikrogravitation ist es daher insbesondere auf und nach Langzeitmissionen von Interesse diese Parameter, inklusive der Muskelsteifigkeit, zu überwachen und Gegenmaßnahmen zu entwickeln. Dabei ist es essenziell, diese Gegenmaßnahmen durch gezielte Muskelfunktionstests zu evaluieren.

In vergangenen Studien wurden dabei bereits mittels „Quick Release“- (QR) und „Sinusoidal Perturbation“ (SP)-Tests die kontraktile und elastischen Eigenschaften des M. triceps surae, der Muskelgruppe die primär für die Plantarflexion und -extension des Fußes verantwortlich ist, vor und nach Raummissionen untersucht. Dabei wurden Änderungen der Muskelsteifigkeit nach dem Aufenthalt der Astronauten in Mikrogravitation festgestellt [13].

Auch in der „Artificial Gravity Bed Rest study with ESA“ (AGBRESA) wurden QR- und SP-Tests angewendet, die Muskelsteifigkeit von Probanden zu überprüfen, welche an 60-tägigen Bettruhestudien teilgenommen haben. Langzeit-Bettruhestudien gelten dabei als probates Mittel, um den Einfluss von Mikrogravitation auf den Körper von Probanden zu simulieren und pathologische Veränderungen nachzubilden [14].

1.2 Zielsetzung und Relevanz - Validität der Messmethoden

Es ist nicht klar, ob die Geräte und Verfahren für durchgeführte Steifigkeitsmessungen ausreichend validiert wurden. Dazu gehört zum einen, ob eine ausreichende Messvalidität gegeben ist, also ob die gewählten Verfahren überhaupt zulässig sind, die Muskelsteifigkeit in diesem Kontext zu erfassen. Auf der anderen Seite ist die Konstruktvalidität unklar. Ist eine ausreichende Unterscheidung zwischen Sehne und Muskel im Messaufbau möglich und zu welchem Zeitpunkt der Bewegung misst man Sehne bzw. Muskel. Auch ist herauszufinden, ob systematische Fehler und Veränderungen erfasst werden können.

Bevor Steifigkeitsmessungen in der Praxis Eingang finden können, muss also die Validität des Quick Release- und des Sinusoidal Perturbation-Test überprüft werden.

Um die Validität dieser Verfahren für die Muskelsteifigkeitsmessung zu überprüfen, soll ein Phantom konstruiert werden, welches die viskoelastischen Eigenschaften des M. triceps surae nachbildet und an welchem mit Hilfe des Muscle Atrophy Research and Exercise System (Abschnitt 2.4) QR- und SP-Versuche durchgeführt werden können. Die so gewonnenen Daten sollen in Bezug auf die muskulotendinöse und muskuloartikuläre Steifigkeit ausgewertet und mit den Daten aus der Bettruhe-Studie AGBRESA verglichen werden. Da die technischen Daten des Phantoms bekannt sind, können mit den erhaltenen Daten aus den Versuchen am SPQR-Phantom die Validität der Messmethoden überprüft werden, indem die Ergebnisse mit den errechneten theoretischen Werten für die Steifigkeiten des Phantoms abgeglichen werden.

1.3 Aufbau der Thesis

Im Folgenden werden in Abschnitt 2 zunächst die für die Arbeit relevanten theoretischen Grundlagen erläutert.

Dabei wird zunächst der aktuelle Stand der Forschung beleuchtet, auf welchem die Arbeit thematisch aufbaut. Es folgt eine Erklärung zum Aufbau und der Funktion des Muscle Atrophy Research And Exercise System sowie eine Beschreibung der Quick Release- und Sinusoidal Perturbation Messmethode. Weiterhin folgt ein Überblick über den Aufbau und alle relevanten Informationen zu den beteiligten Muskeln.

In Abschnitt 3 wird die Methodik beschrieben, welche die Konstruktion und Konzeptionierung des verwendeten Phantoms,, die verwendeten Versuchsprotokolle, sowie die Auswertung der Daten beinhaltet.

Anschließend werden in Kapitel 4 alle Ergebnisse der Messungen am Phantom präsentiert, welche dann schließlich in Kapitel 5 diskutiert werden.

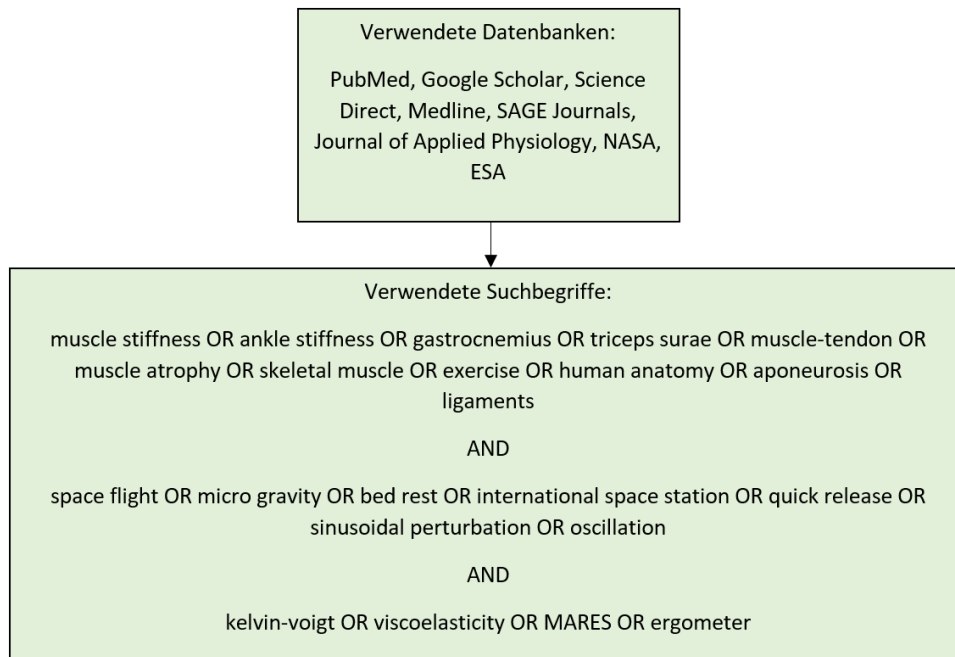
Kapitel 2

Theoretische Grundlagen

In den folgenden Kapiteln werden die Themen behandelt, welche als Hintergrundinformationen zum Verständnis dieser Masterarbeit benötigt werden. Mit dem Ziel, zunächst ein Phantom zu erdenken und zu konstruieren, welches anschließend zur Validierung der Sinusoidal Perturbation (SP)- und Quick Release (QR)-Testmethoden am Muscle Atrophy Research and Exercise System (MARES) verwendet werden kann, werden wir zunächst auf den grundsätzlichen muskulären Aufbau eingehen. Anschließend wird die lineare, mathematisch-technische Näherung von Muskelverhalten behandelt. Diese theoretischen Grundlagen bilden das Grundgerüst zur Bestimmung des methodischen Vorgehens, welches im anschließenden Kapitel näher beschrieben wird.

2.1 Stand der Forschung

2.1.1 Literatursuche



Literaturstellen, die keiner dieser Datenbanken unter Verwendung der genannten Suchbegriffe entstammen, wurden den Literaturverzeichnissen der Quellen entnommen, die über die Suchbegriffe gefunden wurden.

Das Zentrum für Deutsche Luft und Raumfahrt hat folgende Literatur zur Verfügung gestellt:

- Lambertz D et al. (2001) Effects of long-term spaceflight on mechanical properties of muscles in humans
- Lambertz D et al. (2003) Influence of long-term spaceflight on neuromechanical properties of muscles in humans
- Levy E. C. (1959) Complex-curve fitting
- Kearney R. E. et al (1990) System identification of human joint dynamics

2.1.2 Einordnung in den wissenschaftlichen Kontext

Die Skelettmuskulatur hat weitreichende Funktionen im menschlichen Körper. Neben der willentlich gesteuerten Körpermotorik und -statik, ist sie beispielsweise für die Erzeugung von Körperwärme zuständig [15][16]. Neben den technischen Limitationen stellt die pathologische Veränderung der Skelettmuskulatur von Astro- und Kosmoauten eine der großen Hürden der Raumfahrt dar, weswegen die Durchführung von Gegenmaßnahmen und deren Weiterentwicklung essentiell für den Fortschritt in der Raumfahrt ist [17].

Ein Beispiel für pathologische Veränderungen des Muskelapparats ist die Atrophie der Skelettmuskulatur, also eine Verringerung der Muskelmasse. Diese kann verschiedene Ursachen haben. Neben Sarkopenie und neurodegenerativen Erkrankungen, bei denen eine Muskelatrophie als Begleiterscheinung auftreten kann, ist die fehlende mechanische Belastung bei Inaktivität des Muskelapparates zu nennen [18].

Diese fehlende mechanische Belastung und folglich eine Atrophie der Muskulatur tritt klassisch auch bei Langzeitmissionen in der Mikrogravitation des Weltalls auf [4][6][19][13]. Die Reduktion von Muskelmasse und -leistung ist hierbei auch durch aktuell gängige Gegenmaßnahmen nicht vollständig zu verhindern [13][20][21][22].

Die fehlende Belastung hat jedoch nicht nur eine Reduktion der Muskelmasse zu Folge. Auch eine mögliche Veränderung der viskoelastischen Eigenschaften der Muskeln ist schon lange Gegenstand wissenschaftlicher Untersuchungen. In vergangenen Studien wurde bereits gezeigt, dass die Viskoelastizität großen Einfluss auf die Leistung der menschlichen Muskeln hat. Eine wichtige Eigenschaft des elastischen Gewebes ist es beispielsweise, elastische Energie zu speichern und

wieder abzugeben [23]. Hierbei kann elastisches Verhalten im Idealfall wie bei einer Feder durch den Zusammenhang zwischen Kraft und Verformung abgebildet werden [24].

Für eine Evaluierung des viskoelastischen Verhaltens der Muskeln wurden in der Vergangenheit verschiedene Methoden genutzt.

Schon vor über 20 Jahren nutzten Fukashiro et al. eine Ultraschallmethode [25] und Pousson et al. die Quick Release Methode [26] zur Ermittlung des Viskoelastizitätsverhaltens. Hunter und Kearney ermittelten die viskoelastischen Eigenschaften der Muskeln durch die Reaktion der Muskeln auf Sinusoidale Perturbationen [27]. Die letzteren beiden Methoden wurden auch von Lambertz et al. aufgegriffen, um muskulotendinöse sowie muskuluartikuläre Steifigkeit des *M. triceps surae*, vor allem in Bezug auf deren Entwicklung unter Einfluss von Mikrogravitation zu untersuchen. Hierzu wurden Methoden des Quick Releases und der Sinusoidal Perturbation an männlichen Kosmonauten vor und nach 3-6-monatigen Aufenthalten im All durchgeführt.

Die Ergebnisse der ersten Studie zeigten, dass obgleich Gegenmaßnahmen an Bord der ISS durchgeführt wurden, die maximale isometrische Kraft der Probanden um 17 % sank. Die maximale Muskelaktivierung sank in isokinetischen Tests um 39 %. Die muskulotendinöse Steifigkeit, ermittelt durch Quick Release, stieg hingegen um 25 %. Die Steifigkeit des ganzen Gelenks, ermittelt durch Sinusoidal Perturbation, sank unter passiven Bedingungen um 21 %, unter aktiven Bedingungen zeigte sich keine Veränderung (-1 %). Lambertz et al. nahmen an, dass die Erklärung für diese Invarianz ein Adaptionsmechanismus an die Verringerung der passiven Steifigkeit durch eine Erhöhung der aktiven Steifigkeit sei. [13][19]

Weiterhin wurde in der Vergangenheit oftmals ein linearer Zusammenhang der muskulotendinösen Steifigkeit und dem vom Proband ausgeübten Drehmoment festgestellt, bei dem die Steifigkeit der series elastic components (vgl. Abschnitt 2.2.5) mit steigendem ausgeübten Moment steigt [13][28][29][30][31].

2.2 Skelettmuskulatur

Um die Validierung der Testmethoden vornehmen zu können, soll ein Phantom konstruiert werden, welches die realen Bedingungen, also die Nutzung von humanen Probanden, möglichst exakt nachstellt. Es muss also der Vorgang einer Plantarflexion nachempfunden werden. Hierbei ist der Anspruch, das reale Verhalten möglichst genau nachzubilden und dabei die Komplexität des Modells so gering wie möglich zu halten. Im Folgenden wird die Muskulatur des Menschen charakterisiert und hierbei herausgearbeitet, welche Eigenschaften der Muskulatur in einem mathematischen Modell berücksichtigt werden müssen. Hieraus lassen sich Input-Parameter des Modells definieren.

2.2.1 Grundlegende Anatomie des Skelettmuskels

Zum Zwecke dieser Erläuterungen, werden Myofibrillen als die Elementareinheiten der Skelettmuskulatur aufgefasst und die Substrukturen, also Myofilamente und Sarkomere nicht genauer betrachtet. Myofibrillen sind die Bau- und Funktionseinheiten der Muskelfaser und bestehen ihrerseits aus aneinandergereihten Sarkomeren. Mehrere Myofibrillen werden zu einer Muskelfaser zusammengefasst, welche durch das Endomysium, einer Schicht aus Bindegewebe, umhüllt wird. Zwischen den Myofibrillen befinden sich die Mitochondrien des Muskels. Muskelfasern sind üblicherweise zwischen 10 und 120 μm groß und besitzen eine Plasmamembran, das Plasmalemma. Dieses bildet wiederum mit verschiedenen Schichten der Basismembran das Sarkolemma. Angrenzende Muskelfasern werden ihrerseits wieder zusammengefasst und durch das Perymysium, eine robustere Form von Bindegewebe, eingefasst. Dieses Muskelfaserbündel wird als Faszikel bezeichnet und enthält auch die Blutgefäße zur Versorgung des Muskels. Die letzte Organisationsebene ergibt sich durch die Zusammenfassung mehrerer Faszikel, welche ebenfalls durch eine noch stärker Form von Bindegewebe, dem Epimysium, eingehüllt werden. Dieses Bündel bildet das, was gemeinhin als Muskel bezeichnet wird. Die Bindegewebshüllen sind durchzogen von neurovaskulären Strukturen und ihr kollagener Anteil erlaubt die lückenlose Verbindung in Form von Sehnen und Aponeurosen, welche wiederum mit dem Knochen verbunden sind. Auf Sehnen und Aponeurosen wird in folgenden Kapiteln genauer eingegangen. [32]

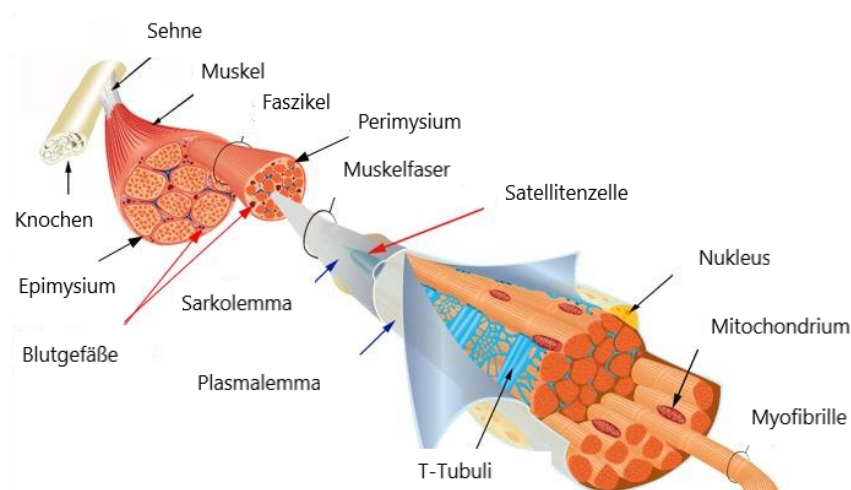


Abbildung 2.1: Muskelaufbau im Querschnitt. Dargestellt sind die verschiedenen Ebenen der Muskelstruktur von der Myofibrille bis zum Muskel, sowie die entsprechenden Bindegewebshüllen und weitere Bestandteile der verschiedenen Muskelebenen. Quelle: modifiziert nach Lonbani [33].

2.2.2 Sehnen

Während Muskeln für die Erzeugung von Kräften zuständig sind, sind Sehnen zur Übertragung dieser Kräfte an Knochen notwendig. Durch diesen Kraftübertragungsweg, können Körper Kräfte ausüben. Sehnen sind uniaxiale Bindegewebsstrukturen, welche zum größten Teil (70-80 %) aus

dicht gepackten, parallel angeordneten Typ 1 Kollagenfibrillen bestehen. In dieser sehnenspezifischen, räumlichen Anordnung liegen die mechanischen Eigenschaften der Sehnen begründet [34]. Auf zellulärer Ebene bestehen Sehnen größtenteils aus Sehnenfibroblasten (Tenozyten), welche für die Sehnenhomöostase und Reparaturmechanismen zuständig sind, sowie zu einem kleineren Teil aus Sehnenstammzellen, welche aufgrund ihrer Fähigkeit sich selbst erneuern und in Tenozyten differenzieren zu können, eine wichtige Rolle in Sehnenwartung und -reparatur spielen. Anatomisch verbunden sind Sehnen und Knochen an der Enthese durch Verschmelzen der Sehne mit dem Periosteum, der dünnen Membran, welche den Knochen umhüllt. Die andere Seite der Sehne verschmilzt an der myotendinösen Verbindungsstelle mit der Faszia des Muskels, welche das Pendant zum Periosteum darstellt. Wie auch die Ultrastruktur des Muskels, sind Sehnen hierarchisch aus fibrillären Strukturen aufgebaut. Sequentiell ergibt sich hierbei die Sehneneinheit aus Faszikeln (Faserbündeln), welche aus Fasern bestehen, welche wiederum aus Fibrillen aufgebaut sind, die wiederum Kollagenmoleküle als Grundbaustein verwenden. Sehneneinheiten sind umschlossen vom Epitenon, welches Reibungen mit angrenzenden Gewebstrukturen reduziert. Alle hierarchischen Strukturen sind parallel angeordnet, woraus sich eine enorme Zugfestigkeit ergibt [35]. Eine schematische Darstellung der Sehnenstruktur findet sich in Abb. 2.2.

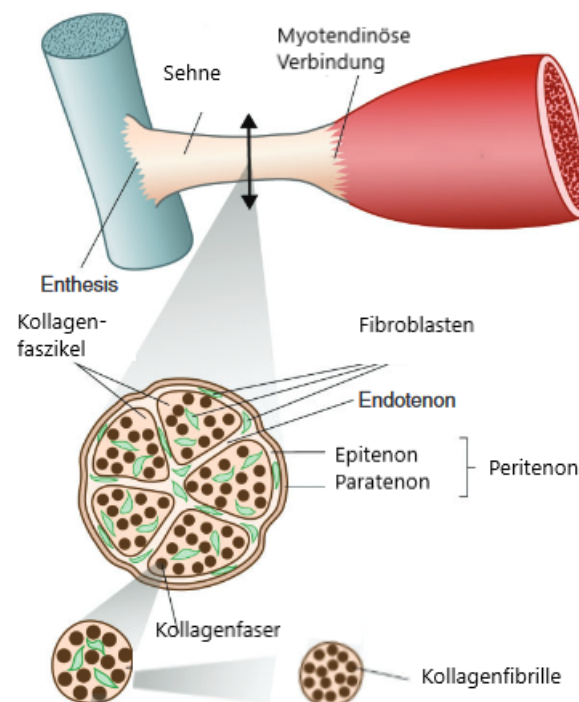


Abbildung 2.2: Schematischer Aufbau der Sehne mit verschiedenen Strukturebenen. Dargestellt sind die verschiedenen funktionellen Sehneneinheiten von Sehne bis Kollagenfibrille, sowie die Ummantelung der Sehne durch das Peritenon. Dieses setzt sich wiederum aus dem Epi- und Paratenon zusammen. Das Epitenon setzt sich auch in die Sehne fort, wo es dann als Endotenon bezeichnet wird. Quelle: modifiziert nach Gaut et al. [34].

2.2.3 Aponeurose

Die Aponeurose ist eine flächige Bindegewebsstruktur, welche den Muskel durchzieht. An ihr setzen einerseits Muskelfasern an und gleichzeitig bildet sie den Übergang zu distalen Sehnen [36]. Im Gegensatz zu Sehnen liegt in Aponeurosen, im Bezug auf die biomechanischen Vorgänge und Eigenschaften, eine deutlich komplexere Situation unter Last vor. Dies ist vor allem in der uniaxialen Belastung von Sehnen begründet, woraus ein berechenbares Verhalten hervorgeht. Die Sehnensteifigkeit nimmt mit der Kraft zu und erreicht dann ein Plateau bei moderaten bis hohen Kräften [37]. Dieses vergleichsweise simple Konzept ist nicht auf Aponeurosen anwendbar. Stand der derzeitigen Erkenntnisse ist, dass das Verhalten von Aponeurosen im Bezug auf Faserverkürzung, Kraftentwicklung und longitudinaler Dehnung durch die Anforderung verkompliziert wird, dass Muskelfasern ein konstantes Volumen während der Kontraktion aufweisen müssen [38][39]. Der Donnan-Effekt und osmotische Equilibria zwingen den Muskel, sich rechtwinkling zur Faserverkürzungsrichtung auszubreiten. Diese transversalen Kräfte führen zur Dehnung der Aponeurose und im Umkehrschluss zur Reduktion der Dehnung in longitudinaler Richtung [40][41]. Es wird vermutet, dass diese transversale Dehnung der Aponeurose durch das Endo-, Peri- bzw. Epimysium ermöglicht wird. Dieses wird durch die Aktivierung der Muskelfasern und Steigerung von Flüssigkeitsdrücken innerhalb und außerhalb des Muskels, bedingt durch Kontraktionen, gedehnt, was sich wiederum auf die Aponeurose überträgt [42]. Die Dehnung der Aponeurose in longitudinaler Richtung scheint auf dem gleichen Mechanismus zu beruhen [43]. Im Falle des Endo- und Perimysiums konnte gezeigt werden, dass die Kollagenfasern insofern in ihrer Geometrie variiert werden können, dass sie einer radialen Ausbreitung der Muskelfasern bei kurzen Längen und einer longitudinalen Dehnung bei großen Längen entgegenwirken können [44]. Die Kollagenfasern des Epimysiums sind wiederum mit den Kollagenfasern des Perimysiums verbunden und sind bereits als Teil der Sehne zu betrachten [45]. Es ist also eine durchgehende Kraftübertragungskette von Muskelfasern, Bindegewebsstrukturen bis zu Sehnen erkennbar. Ein weiterer Aspekt sind Kollagenfasern, die am Ende von Muskelfasern ansetzen und ihrerseits Kräfte an Sehnen übertragen können. Die kollagenen Fasern von Aponeurose und am Ende von Muskeln greifen hierbei ineinander [46]. Die Summe aller Effekte führt dazu, dass die Aponeurose höheren transversalen Dehnungen ausgesetzt ist als freie Sehnen. Dies wiederum hat Einfluss auf die Steifigkeit der Aponeurose. Entsprechende Studien konnten zeigen, dass eine breitere Aponeurose zu zwei bis drei mal größerer longitudinaler Steifigkeit führt, im Gegensatz zu passiver Dehnung der Aponeurose [47]. Hieraus ist abzuleiten, dass die longitudinale Steifigkeit der Aponeurose von den kontraktiven Gegebenheiten abhängt. Eine weitere Konsequenz ist, dass die Menge an Energie die unter bestimmter Last gespeichert und wieder abgegeben werden kann, nicht fix ist. Die einfache Schlussfolgerung, dass Muskel, Aponeurose und freie Sehne lediglich als Reihenschaltung zu betrachten ist und aus dieser die Energierückgewinnung aus passiven Elementen entsprechend errechnet werden kann, greift zu kurz. Durch die Komplexität der Aponeurose und der dynamischen Steifigkeit, ist die Situation deutlich komplizierter [48]. Eine schematische Darstellung der Lage der Aponeurosen findet sich in Abbildung 2.3.

2.2.4 *Musculus triceps surae*

Der *Musculus triceps surae* ist eine Muskelgruppe bestehend aus *M. gastrocnemius*, *M. soleus* und *M. plantaris*. Die Verbindung von Gastrocnemius und Soleus befindet sich an der Achillessehne, der stärksten und dicksten Sehne des menschlichen Körpers [26]. In Bezug auf das Sprunggelenk wirken Gastrocnemius und Soleus als Synergisten. Der Plantaris existiert nur etwa bei 90 % der Bevölkerung [27] und strahlt mit seinen Fasern in die Faszien des Soleus und die Achillessehne ein. Er hat nur eine sehr geringe funktionelle Bedeutung.

Der *M. gastrocnemius* besteht aus zwei Muskelköpfen, *caput mediale* (innerer Muskelkopf) und *caput laterale* (seitlicher Muskelkopf), welche beide am unteren Teil des Oberschenkelknochens entspringen. Zu seinen Funktionen gehört die Beugung des Fußes, die Beugung des Kniegelenks sowie die Supination des Fußes.

Obgleich der *M. soleus* direkt unter dem Gastrocnemius liegt und größtenteils von diesem bedeckt wird, sind beide Muskeln voneinander getrennt und besitzen eine meist getrennte Aponeurose (Abb. 2.3). Der Soleus setzt sowohl an Tibia als auch an Fibula an und seine Funktion ist die Plantarflexion sowie die Supination des Fußes.

Die gemeinsame Sehne von Gastrocnemius und Soleus heftet am Fersenbein an. Der so entstehende Hebel kann Kraft auf das Sprunggelenk übertragen, die entstehende Kraft hängt dabei von Abstand des Anheftpunktes der Achillessehne zur Sprunggelenksachse ab. [26]

Der *M. triceps surae* ist primär für die Plantarflexion verantwortlich. Andere Plantarflexoren wie der *M. tibialis posterior* und die *Musculi peronei* haben nur sehr kurze kontraktile Fasern [29].

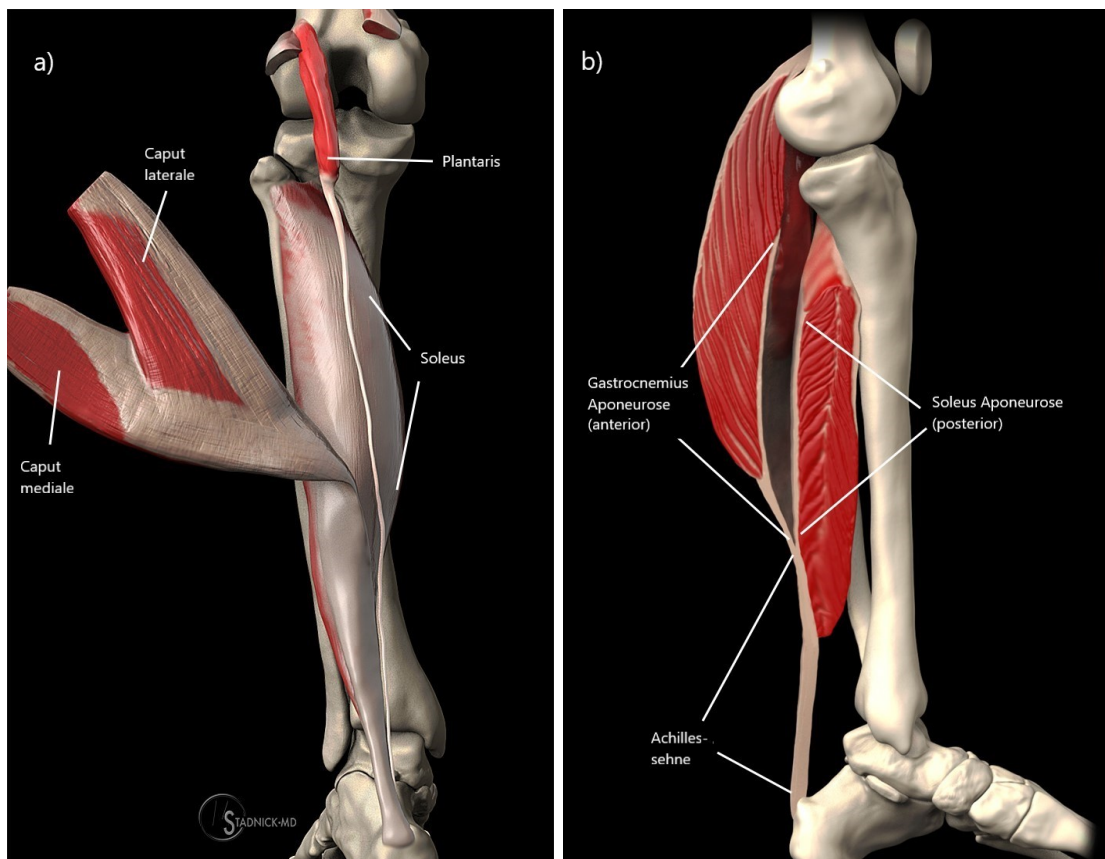


Abbildung 2.3: Schematische Darstellung des *M. triceps surae*, bestehend aus a) den Muskeln sowie b) der aus den beiden Aponeurosen fusionierten Achillessehne. Quelle: modifiziert nach Bodor [49].

2.2.5 Mechanische Eigenschaften des Muskel-Sehnen-Komplexes

Muskel-Sehnen-Komplexe sind verantwortlich für Bewegungen an den Gelenken und ihre mechanischen Eigenschaften sind in erheblichem Maße am Prozess der Energiespeicherung und -rückgewinnung, sowie an dem damit verbundenen Prozess der Übertragung von Kräften beteiligt [50][51]. Die Eigenschaften der Muskel-Sehnen-Komplexe teilen sich dabei in kontraktile und viskoelastische Eigenschaften. Nach Hills Modell der Muskelmechanik [52] sind im skeletalen Muskel-Sehnen-Komplex elastische Elemente des Muskels in Serie mit den kontraktilen Elementen des Muskels geschaltet. Die elastischen Elemente werden daher als “series elastic components” (SEC) bezeichnet. Die SEC beinhalten dabei wiederum aktive und passive Strukturen [53]. Das meistverwendete Maß zur Charakterisierung der mechanischen Eigenschaften ist die Steifigkeit, diese kann z.B. über QR- und SP-Tests bestimmt werden [54][55]. Mit Steifigkeit ist der Widerstand gemeint, den ein System einer Perturbation von außen entgegenbringt [52]. Unter Verwendung eines an *in vivo* Messungen angepassten Hill-Modells [56] kann diese Steifigkeit in Relation zur Steifigkeit des Series Elastic Component (SEC) gesetzt werden [57][58]. Die Steifigkeit der SEC wird auch als “short-range”-Steifigkeit bezeichnet, da sie bei kleinen, rapiden

Bewegungen der gesamten Muskelsteifigkeit gleicht [59]. Die SEC passen ihre Eigenschaften sehr stark an den funktionellen Bedarf an [60]. Unter Bedingungen der Mikrogravitation sinkt dieser Bedarf sehr stark und es kommt ggf. zu erheblichen Veränderungen in der SEC-Struktur. Diese Veränderung wird über u.a. Sinusoidal Perturbation und Quick Release Methoden charakterisiert.

2.3 Artificial Gravity Bed Rest study with ESA

Im Jahr 2019 wurde die 60-tägige Bettruhestudie Artificial Gravity Bed Rest study with European Space Agency (AGBRESA) am DLR durchgeführt. Das Ziel der AGBRESA Studie ist es, die Effizienz von künstlicher Gravitation in Form von Zentrifugation in einer Kurzarm-Zentrifuge als Gegenmaßnahme gegen die negativen Effekte von simulierter Mikrogravitation während einer 60-tägigen Bettruhe Studie zu untersuchen. In zwei Kampagnen sind je 12 Probanden einer 60-tägigen Bettruhe unter einem -6° Neigungswinkel des Betts ausgesetzt worden.

Der -6° Neigungswinkel simuliert hierbei die Verschiebung des Flüssigkeitsvolumens in die oberen Körperareale. Die generelle Bettruhe simuliert durch den einhergehenden Bewegungsmangel die gewünschte Nichtbelastung der unteren Extremitäten.

Dabei wurden drei verschiedene experimentelle Konditionen, denen die Probanden als Gegenmaßnahme ausgesetzt wurden, untersucht: 30 Minuten kontinuierliche künstliche Gravitation jeden Tag über den Zeitraum der Bettruhe, periodische Gravitation bestehend aus fünf Intervallen mit jeweils sechs Minuten Gravitation und dreiminütigen Pausen sowie einer Kontrollgruppe ohne den Einsatz der Kurzarm-Zentrifuge. AGBRESA beinhaltet dabei mehrere Unterstudien, welche verschiedene Effekte der Bettruhe auf den menschlichen Körper untersuchen sollen. Eine dieser Studien untersucht Veränderungen der elastischen Eigenschaften des Muskel-Sehnen Apparats mit Hilfe von Quick Release und Sinusoidal Perturbation Tests am Sprung- und Kniegelenk vor und nach der 60-tägigen Bettruhe. [61]

Die Ergebnisse der AGBRESA Studie werden zum aktuellen Zeitpunkt ausgewertet und evaluiert und werden bezüglich der Quick Release und Sinusoidal Perturbation Methode in dieser Arbeit exemplarisch mit den selbst generierten Daten des konstruierten Phantoms verglichen.

2.4 Muscle Atrophy Research and Exercise System

Das Muscle Atrophy Research and Exercise System (MARES) ist ein multifunktionales, computergestütztes Dynamometer zur neuromuskulären, muskuloskelettalen und biomechanischen Forschung.

Das System wurde von der European Space Agency (ESA) mit dem Ziel entwickelt, den Effekt von Mikrogravitation auf das Muskel-Skelett-System besser zu verstehen. [62]

Dabei bietet das MARES ebenfalls die Möglichkeit Trainingsprotokolle durchzuführen und da-

mit mögliche Gegenmaßnahmen zu den in Mikrogravitation auftretenden Effekten, im Speziellen der Atrophie von Muskeln, zu evaluieren [63].

Die drei Hauptziele des MARES sind damit [64]:

- Forschung an muskuloskeletaler, biomechanischer, neuromuskulärer und neurologischer Physiologie
- Untersuchung der Effekte von Mikrogravitation auf den menschlichen Körper
- Evaluation von Gegenmaßnahmen gegen diese Effekte und gleichzeitige Beurteilung der Effektivität von Übungsprotokollen

Im Rahmen der Space Shuttle Mission STS-131 (Space Transportation System) wurde am 05. April 2010 ein Muscle Atrophy Research and Exercise System auf der Raumfähre „Discovery“ mit weiterer Fracht zur Internationalen Raumstation (ISS) geflogen und in einem „International Standard Payload Rack“-Forschungsmodul (ISPR) des Columbus-Raumlabor installiert. Es ist ein Teil der Human Research Facility (HRF) [65]. Abbildung 2.4 zeigt das MARES auf der Internationalen Raumstation.

Neben der Untersuchung des muskuloskelettalen Systems, hilft das MARES ebenfalls dabei, die Gesundheit der ISS-Crew zu dokumentieren und zu erhalten [65]. Eine reliable Methode zur Erfassung der Änderung der Muskelkraft während Raummissionen, und nicht erst nach der Rückkehr zur Erde, war vor Entwicklung des MARES nicht vorhanden. Entsprechend wurde die Retest-Reliabilität des MARES vor dem Start der Raummission STS-131 überprüft und verifiziert [66].



Abbildung 2.4: Das Muscle Atrophy Research and Exercise System an Bord der ISS mit ESA Astronautin Samantha Cristoforetti. Quelle: Logbuch der ISS-Astronautin Samantha Cristoforetti [67].

Die Hardware des Mares besteht grundlegend aus fünf Komponenten:

Die elektro-mechanische Hauptbox

Die Hauptbox ist das Herzstück des MARES und beinhaltet den Motor, welcher Drehmomente zwischen 3 Nm und 900 Nm generieren und Winkelgeschwindigkeiten zwischen 5°/sec und 515°/sec erreichen kann. Weitere Komponenten der Hauptbox sind die zum Motor gehörende Elektronik, wie Spannungs-, Steuerungs- und Servoeinheiten, Lüfter und Anschlüsse, um das MARES mit der HRF und weiteren externen Geräten zu verbinden.

Ein Fixierungssystem für den Probanden

Das Fixierungssystem besteht aus einem frei verstellbaren Sitz für den Probanden sowie zugehörigen Handgriffen, Hebeln, Gurten und Polstern zur Fixierung der verschiedenen Körperteile. Diese Komponenten erlauben Konfigurationen für Probanden vom 5. bis zum 95. Perzentil. Der Sitz kann dabei durch Rotation und Translation eines Pantograph-Armes in verschiedene Positionen relativ zur Hauptbox gebracht werden. Ziel des Fixierungssystems ist es, Muskelgruppen gezielt isoliert zu betrachten und gleichzeitig eine Übereinstimmung der Gelenkachsen mit der Motorachse des MARES zu gewährleisten. Dies ist relevant, da eine Verschiebung dieser Achsen zu einem Fehler in Messungen führt, der größer wird, je weiter die Achsen voneinander abweichen [63][68]. Gleichzeitig soll während der Messungen ein möglichst hoher Komfort für den Probanden erreicht werden.

Ein linearer Adapter

Der lineare Adapter übersetzt die rotatorische Bewegung des Motors in eine lineare Bewegung und ermöglicht Trainingseinheiten in verschiedenen Neigungen und Positionen. Dadurch kann eine große Vielfalt von Übungen abgedeckt werden. Der Adapter beinhaltet außerdem Kraft- und Momentsensoren an den Handgriffen des MARES.

Ein Vibration-Isolations Frame

Das Vibration-Isolations Frame minimiert den Einfluss von Vibrationen auf die Messungen und isoliert das MARES mechanisch vom ISPR und der ISS.

Ein Laptop

Zuletzt benötigt das MARES einen Laptop, welcher die Steuerungssoftware enthält und als User Interface dient. Mit diesem können die Operationen des MARES von der ISS-Crew bedient und gesteuert werden. Die Software ist dabei so konzipiert, dass sie den Bediener mit klaren, linearen Strukturen durch alle Schritte des Testablaufs führt. Die Software ist komplett programmierbar und erlaubt so eine komplexe Gestaltung der Test-

und Trainingsabläufe. Dabei wird aus einer Basis an grundlegenden Kontrollalgorithmen, den „basic motion units“ (BMUs), für den Motor ausgewählt. Es können Benutzerprofile mit verschiedenen Einstellungen für unterschiedliche Probanden und Testabläufe erstellt werden. [69]

Am MARES kann von Astronauten und Probanden eine Vielzahl verschiedener physischer Aktivitäten durchgeführt werden. Dabei erlaubt es die Aufzeichnung von Kräften und Drehmomenten isolierter Muskeln in der Nähe von Gelenken durch die Messung von Verhältnissen zwischen Position, Geschwindigkeit, Kraft oder Drehmoment als Funktion der Zeit. Die Abtastfrequenz des MARES beträgt dabei 2000 Hz.

Weiterhin unterstützt das System Versuche an sieben verschiedenen Gelenken der jeweils linken oder rechten Körperhälfte. Dazu gehören das Sprung-, Knie-, Schulter-, Ellenbogen- und Handgelenk sowie die Hüfte und der Rumpf an denen Flexions- oder Extensionsbewegungen durchgeführt und aufgezeichnet werden. Zusätzlich können durch den linearen Adapter translatorische Bewegungen durchgeführt und Beine und Arme für Messungen oder Trainingsprogramme durch eine Pressbewegung linear belastet werden. Auch mehrgelenkige Tätigkeiten sind möglich [62]. Zudem unterstützt das MARES sowohl isometrische, isotonische und isokinetisch Übungen, bei denen auch die Parameter der Bewegungen wie Geschwindigkeit, Drehmoment, Kraft und Leistung gesteuert werden können. Dabei sind die Übungen frei kombinierbar und es können individuell abgestimmte Übungen und Abläufe durchgeführt werden.

Weitere Verfahren, die mit dem MARES durchgeführt werden können, sind der sogenannte „Quick Release“ sowie der „Sinusoidal Perturbation“-Test (vgl. Abschnitt 2.6 und 2.7). Diese Testabläufe werden eingesetzt, um die Muskelsteifigkeit zu überprüfen [13]. Ein mögliches Einsatzgebiet ist die Überprüfung der Muskelsteifigkeit vor- und nach Einfluss von Mikrogravitation auf Raummissionen.

Ein Muscle Atrophy Research and Exercise System steht zum aktuellen Zeitpunkt im :envihab Labor des DLR Standortes in Köln und wurde für die in dieser Arbeit durchgeführten Messungen genutzt.

Abbildung 2.5 zeigt den für die Messung des rechten Sprunggelenks konfigurierten Aufbau. Hier können Flexions- und Extensionsbewegungen durchgeführt und aufgezeichnet werden. Dieser Aufbau des MARES wird bei QR- sowie SP-Messungen benutzt.

Direkt an der Drehachse ist der „short lever - Arm“ (SLA, Abbildung 2.5a) befestigt, der die Bewegungen und Kräfte auf die Achse des MARES überträgt. Am SLA befindet sich die Fußhalterung, in welcher der Fuß des Probanden bei den Messungen fixiert wird. Ebenfalls auf Höhe der Achse befestigt, ohne aber Kräfte auf diese auszuüben, ist der static lever. Dieser kann für verschiedene Aufbauten gedreht, positioniert und fest eingerastet werden und bildet damit die Grundlage, um verschiedene MARES-Bauteile an ihm zu befestigen. Dazu gehören in diesem Aufbau die Kissen, welche bei QR- und SP-Messungen den Oberschenkel einfassen. Rechts in Abbildung 2.5 ist der Sitz zu sehen, in dem der Proband während der Messungen fixiert ist.

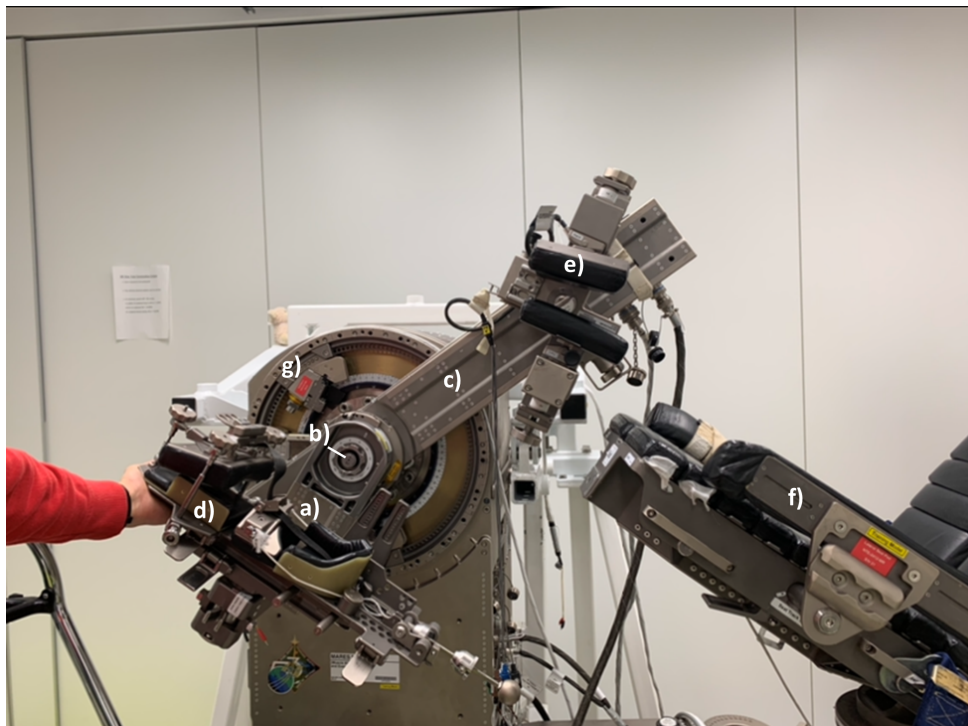


Abbildung 2.5: Das Muscle Atrophy Research And Exercise System. Der gezeigte Aufbau ist für eine Messungen der Flexion/Extension des rechten Sprunggelenks konfiguriert. a) short lever, b) Drehachse, c) static lever, d) Fußhalterung, e) Kissen zur Fixierung des Oberschenkels, f) Sitz des Probanden, g) oberer Endstop. Quelle: eigene Fotografie.

An der Hauptbox befinden sich die beiden sogenannten Endstops (Abbildung 2.5g). Diese grenzen den Bewegungsspielraum des Systems um die Achse ein und befinden sich an den Enden der Range of Motion des Probanden (ROM). Die Software des MARES bremst kurz vor Erreichen der Endstops automatisch ab. Sollten die Endstops dennoch berührt werden, beendet das System den Testlauf. Weiterhin besitzt das MARES die Möglichkeit, das Trägheitsmoment der sich an der Achse befindlichen Bauteile in der Software voreinzustellen. Das Trägheitsmoment des SLA kombiniert mit dem Fußteil, wie es im Laufe der AGBRESA Studie für QR und SP Messungen genutzt wurde, beträgt $0,35 \text{ kgm}^2$. Die genauen technischen Spezifikationen des MARES befinden sich im Anhang [70].

2.5 Methoden zur Charakterisierung von Gelenkdynamik

Wie im vorangegangenen Kapitel beschrieben, ergibt sich das muskulotendinöse und musku-loartikuläre Systemverhalten aus einer Reihe von Untersystemen, die wiederum miteinander in Beziehung stehen. Technisch direkt beobachtbar ist hierbei nur Position und Drehmoment des Gesamtsystems. Indirekte Informationen über das System sind noch über Elektrische Myografie (EMG)-Messungen möglich. Die direkte Beobachtung der beteiligten Untersysteme ist nur schwer möglich, weshalb sich ein klassisches Problem der Systemidentifikation ergibt, wenn, wie

in unserem Falle, eine mathematische Modellierung des Systems gewünscht ist. Im Rahmen der Systemidentifikation muss unter Nutzung der Beziehung zwischen Input (Position oder Drehmoment) und Output (Drehmoment oder Position) ein Modell zur Beschreibung des Systems identifiziert werden. Es können Modelltypen nach unterschiedlichem Detailreichtum der quantitativen Aussagekraft unterschieden werden. Zu unserem Zwecke wird ein mathematisches Modell angestrebt, welches den größten Detailreichtum bietet, da sie in der Lage sind das Systemverhalten unter relativ allgemeinen Bedingungen vorherzusagen. Mathematische Modelle werden in parametrische und nicht-parametrische unterschieden. Zur Durchführung des Ziels dieser Arbeit ist ein parametrisches Modell zwingend notwendig. Im Gegensatz zu nicht-parametrischen Modellen wird das Systemverhalten durch relativ wenige Parameter beschrieben. Dies setzt allerdings eine sehr akkurate Modellstruktur voraus, um sicherzugehen, dass alle relevanten Inputs berücksichtigt sind und gleichzeitig möglichst keine Überschneidungen der Inputparameter vorliegen.

Außerdem muss nun die experimentelle Herangehensweise spezifiziert werden. Hierbei kann ohne expliziten Input, mit Drehmomentinput oder mit Positionsinput gearbeitet werden. Im vorliegenden Falle wird versuchsbedingt ein Positionsinput (SP) bzw. ein Drehmomentinput (QR) gewählt. Dies bedeutet, dass für SP auf die Position des Gelenks Einfluss genommen wird und die resultierende Drehmomentantwort erfasst wird. Bei QR wird ein Drehmoment angelegt und die resultierende Positionsantwort inklusive Geschwindigkeit und Beschleunigung erfasst.

Weiterhin ist die Form des Inputs von Bedeutung. Es werden transiente, sinusoidale und stochastische Inputs unterschieden. Theoretisch sollten alle Formen zum gleichen Ergebnis führen, es ergeben sich jedoch in der Praxis Vor- und Nachteile.

Kearney et al. haben in ihrem Review die wichtigsten Messmethoden inklusive der Vor- und Nachteile dargestellt [71].

2.6 Quick Release

Der Quick Release ist eine Testmethodik zur Bestimmung der elastischen Eigenschaften der series elastic components eines isolierten Muskels. Der Name Quick Release ergibt sich aus dem schnellen Lösen einer isometrischen Kraft, die zuvor gegen den zu messenden Muskel wirkt.

Der QR-Test kann an verschiedensten Muskeln des menschlichen Körpers angewendet werden. Im Folgenden wird die gängige und für diese Arbeit relevante Durchführung des Quick Release zur Vermessung der Plantarflexoren des Muskelkomplexes *M. triceps surae* beschrieben. Die Beschreibung bezieht sich dabei primär auf die Durchführung des Quick Release durch das Muscle Atrophy Research and Exercise System, da sich die Arbeit auf diese Durchführung des Quick Releases stützt, grundlegend ist der Quick Release jedoch nicht MARES-exklusiv und kann auch mit anderen Dynamometern durchgeführt werden.

Der Proband befindet sich hierbei im MARES (Abbildung 2.5) und ist in sitzender Position

festgeschnallt. Die Hüfte befindet sich in einem 90° Winkel, das Kniegelenk variiert je nach Testreihe zwischen 90 und 120° . Der Fuß des Probanden ist auf einer Platte befestigt, welche über den SLA direkt mit der Drehachse des Motors des MARES verbunden ist. Die Position des Fußes ist hierbei so arretiert, dass die Rotationsachse des MARES mit der horizontalen Bimalleolarachse des Fußgelenkes übereinstimmt. Die neutrale Position (NP) des Fußes liegt bei 90° , sie kann aber zur Erzeugung verschiedener Vorspannungen des Muskels auf verschiedene Startpositionen eingestellt werden.

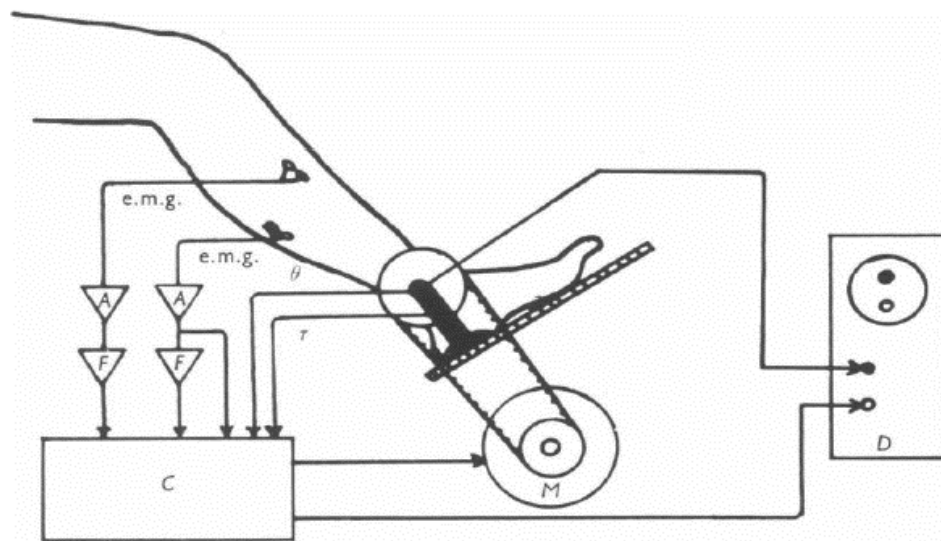


Abbildung 2.6: Fußergometer zur Messung der Muskeldynamik bei Plantarflexion und -extension nach G. C. Argawal und G. L. Gottlieb. Quelle: Agarwal et al. [54].

Dieser Messaufbau gewährleistet eine isolierte Krafterzeugung durch den *M. triceps surae* (Abschnitt 2.2.4), bestehend aus *M. soleus* und *M. gastrocnemius* und ein Einwirken der Oberschenkel- und Hüftmuskulatur wird minimiert.

Zu Beginn des Tests wird die MVC des Unterschenkels bestimmt. Dies geschieht über die Funktion Angle Torque Relation (ATR) des MARES. Der Motor des MARES gibt hierzu eine isometrische Kraft auf, gegen die der Proband mit dem Vorfuß so stark wie möglich drücken und einige Sekunden halten soll. Mit Hilfe von Dehnungsmessstreifen (DMS) als Drehmomentsensoren kann so die MVC bestimmt werden. Diese Messung wird jeweils drei Mal durchgeführt und der Mittelwert aus den drei jeweils höchsten erreichten Kräften gebildet. Die Messung der MVC erfolgt mehrfach für verschiedene Winkel der Fußplatte und somit bei verschiedenen Winkeln zwischen Fußsohle und Unterschenkel (80 bis 100° in 10° -Schritten). Die Anzahl der Winkel mit denen die ATR durchgeführt wird, kann durch eine engere Positionierung der Endstops des MARES angepasst werden. Winkel die außerhalb der Endstoppositionen liegen, werden nicht mehr vom MARES berücksichtigt.

Um einen Quick Release durchzuführen, wird mit der Fußspitze im Sinne einer Plantarflexion

eine isometrische Kontraktion erzeugt und eine Kraft auf die Fußplatte aufgebaut. Dabei gibt das MARES einen Drehmoment vor, welches der Proband für einige Sekunden halten muss. Diese vorgegebenen Drehmomente sind prozentuale Abstufungen des MVC. Dabei werden Kräfte von 25, 50 und 75 % der Maximalkraft (25 %MVC, 50 %MVC, 75 %MVC) vorgegeben. Diese Abstufung der Maximalkraft gewährleistet die Durchführung bei verschiedenen Drehmomenten. Die Reihenfolge, mit der der Proband die verschiedenen Abstufungen der Maximalkraft ausüben muss, ist randomisiert. Um die isometrische Kontraktion auf dem gleichen Level zu halten, wird ein visuelles Feedback ausgegeben, welches für Proband und Testleiter zu sehen ist.

Die Achse und damit die Fußplatte sind, während der Proband Kraft aufbringt, zunächst unbeweglich. Nach Halten der vorgegebenen Kraft löst sich die vom Motor produzierte Gegenkraft schlagartig und die Fußplatte gibt nach. Der Zeitpunkt des Auslösens ist dem Probanden unbekannt. Der Fuß plantarflexiert bis zum physiologischen Ende der Range of Motion (ROM), also des maximal möglichen Bewegungsspielraums.

In jeder Testreihe werden für jeden MVC Bereich (25 %MVC, 50 %MVC, 75 %MVC) jeweils drei Wiederholungen durchgeführt. Es werden also neun einzelne Quick Releases pro QR-Testreihe durchgeführt. Durch Erholungspausen zwischen den Quick Release Tests wird eine Ermüdung der Muskeln vorgebeugt.

Das MARES zeichnet über die komplette Laufzeit durchgehend die Verläufe verschiedener Parameter auf. Die Charakteristik der SEC werden am Anfang der Bewegung gemessen, wenn eine anfängliche Abnahme der Beschleunigung durch die SEC auftritt. Das Verhältnis der Variation von Beschleunigung sowie Winkelposition wird als Merkmal für die Steifigkeit der SEC genutzt. Die Trägheit des Systems wird über die maximale Winkelbeschleunigung und das aufgebrachte Drehmoment errechnet (vgl. Formel 3.17). Aus der Änderung der Winkelbeschleunigung, der Änderung der Winkelposition des lever arms und damit der Fußplatte sowie dem errechneten Trägheitsmoment kann anschließend die muskulotendinöse Steifigkeit der SEC der Plantarflexoren berechnet werden. Eine genauere Beschreibung der Berechnung der Steifigkeit nach den Messparametern findet sich in Abschnitt 3.6.1.

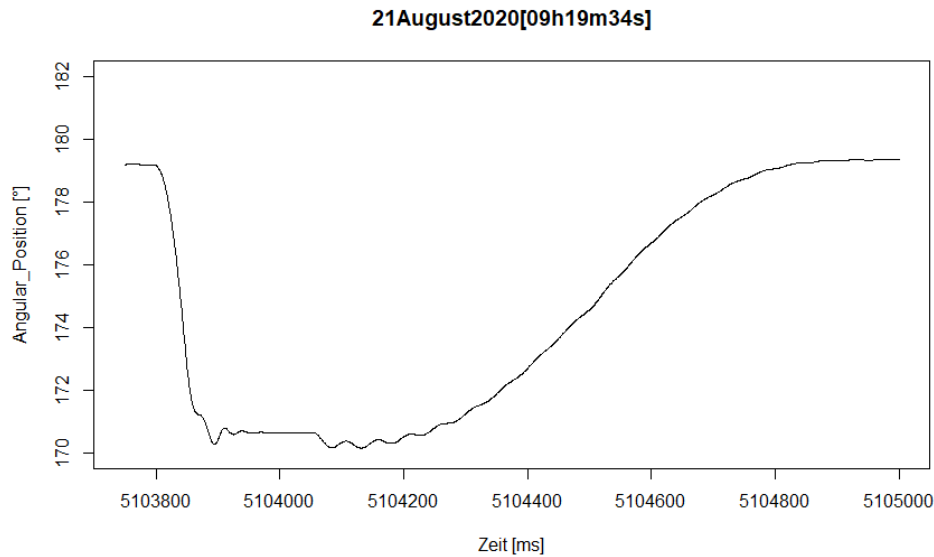


Abbildung 2.7: Klassischer Winkelverlauf eines Quick Releases aufgetragen über der Zeit nach Messbeginn Quelle: Eigene Messung vom 21. August 2020.

In Abbildung 2.7 ist die Winkelposition des MARES direkt nach dem Auslösen eines Quick Releases zu sehen. Der SLA bewegt sich in Negativrichtung und verringert den Positionswinkel bis zum Ende der ROM. Anschließend kehrt die Fußplatte mit einer geringeren Beschleunigung wie bei Auslösen des QRs in die Ausgangsposition zurück. Der vorliegende QR entstammt einer in dieser Arbeit durchgeführten Messreihe vom 21. August 2020.

2.7 Sinusoidal Perturbation

Für die Testung im Rahmen der SP-Tests, werden sinusoidale Inputs verwendet. Diese Methodik birgt einige signifikante Vorteile. Diese umfassen unter anderem:

- Der Input ist auf diskrete Frequenzen konzentriert
- Rauschunterdrückung möglich durch Mittelung mehrerer Messdurchläufe
- Nichtlinearitäten des Systems sind vergleichsweise einfach zu detektieren
- Einfache Übertragung der nichtparametrischen Frequenzantwort in eine parametrische Transferfunktion (z.B. Bode-Plot)
- Vergleichsweise simpler Aufbau und direkte Analysemöglichkeiten

Doch auch Nachteile sind vorhanden, dies sind beispielsweise:

- Langwierige Datenakquise, mit der Konsequenz, dass eine vergleichsweise geringe Menge an Frequenzen gemessen wird, wodurch die Auflösung leidet
- Alle Messparameter außer der zu variierenden Frequenz müssen sehr stabil gehalten werden, was im Bezug auf menschliche Probanden schwierig sein kann (z.B. auf Grund von Muskelermüdung und/oder Reflexantworten)

Praktisch bedeutet dies, dass eine SP-Testung an Humanprobanden folgendermaßen abläuft: Der Proband wird in sitzender Position in einem entsprechenden Sitz fixiert, sodass sich Oberschenkel und Oberkörper in einem 90° -Winkel zueinander befinden. Das Knie wird in 120° -flekterter Position auf einer Ablage positioniert. Der Fuß wird in Neutralstellung (90° -Winkel zwischen Unterschenkel und Fuß) in einer entsprechenden Aufnahme festgeschnallt. In dieser Position wird nun einmalig die Maximum Voluntary Contraction (MVC) bestimmt, indem der Proband im Sinne einer Plantarflexion mit größter Kraft gegen die Fußaufnahme drückt, welche das resultierende Drehmoment erfasst. In der eigentlichen SP-Testung ist der Proband nun aufgefordert, entsprechend der Anweisung durch den SP-Test, Kraft prozentual der MVC aufzubringen, während die Fußaufnahme durch einen Aktuator in einer sinusoidalen Form bewegt wird, also wippt. Die Parameter der SP sind wie folgt: Es wird dem Sprunggelenk für 4s eine sinusoidale Oszillation mit einer Peak-to-Peak Amplitude von 3° um die Neutralposition aufgeprägt. Hierbei werden in Einzelmessungen Frequenzen von 4 bis 16 Hz abgedeckt, wobei bis 12 Hz Inkremente von 1 Hz verwendet werden und bei und über 12 Hz Inkremente von 2 Hz. Während dieser 4s gilt es 25 %, 50 %, 0 % und schließlich 75 % MVC beizubehalten. Das Erreichen dieser Drehmomentvorgabe durch den Probanden wird durch die Echtzeit-Darstellung des Tiefpass-gefilterten Drehmomentverlaufs gewährleistet. Zwischen den Durchläufen pro Frequenz befinden sich Pausen von ca. 20s Länge, in denen der Proband sich entspannen kann. Aus den gemessenen Werten für Winkelposition und dem entsprechenden Drehmoment werden Frequenzantworten generiert und das Amplitudenverhältnis (Gain) und die Phasenverschiebung (Phase) werden über die eingestellte Frequenz aufgetragen. Die so entstandene Visualisierung nennt sich Bode-Plot und dient der Darstellung des frequenzabhängigen Verhaltens eines linearen, zeitinvarianten Systems. Die nachstehende Abbildung zeigt einen Bode-Plot, welcher die Ergebnisse einer Testung von humanen Gelenkdynamiken darstellt. Diese sind, anders als hier thematisiert, über einen stochastischen Positionsinput generiert worden, die Resultate weisen jedoch, wie bereits angesprochen, keine Unterschiede zu einer Ermittlung der Ergebnisse mit Hilfe einer SP-Testung auf. Im weiteren Verlauf dieses Kapitels wird auch noch ein beispielhafter Gain-Verlauf für eine SP-Testung thematisiert.

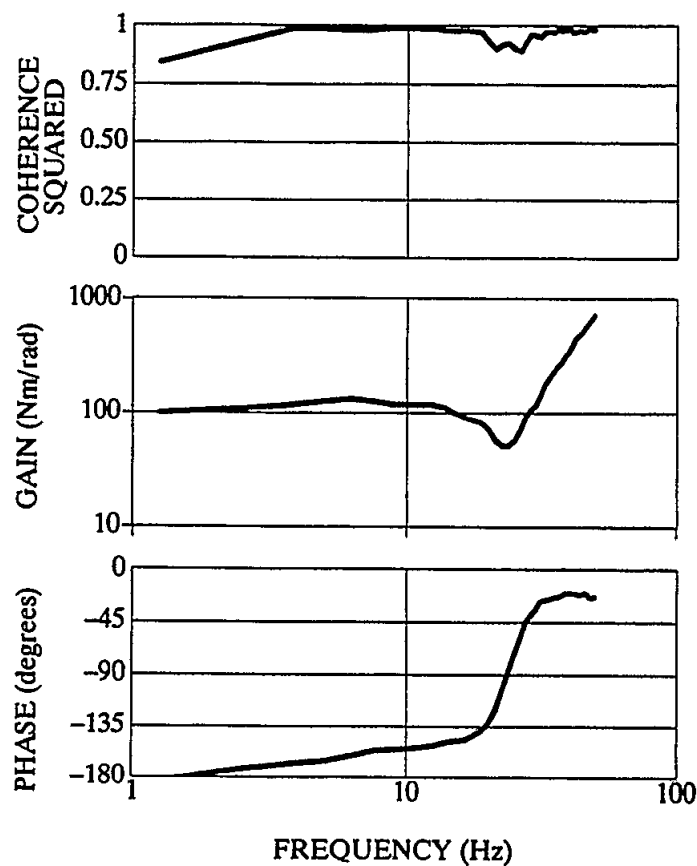


Abbildung 2.8: Ergebnisse einer linearen Übertragungsfunktionsanalyse für humane Sprunggelenkdynamik, also die Gelenksteifigkeit in Abhängigkeit der Frequenz. Die Datengrundlage bildet ein stochastischer Positionsinput bei Frequenzen zwischen 1 und 50 Hz. Dargestellt ist die Kohärenz des Modells mit den Daten, sowie Gain und Phase in Abhängigkeit der Frequenz. Gain und Frequenz sind hierbei logarithmisch dargestellt. Quelle: Kearney et al. [71].

Die Abbildung 2.8 zeigt die Ergebnisse einer linearen Übertragungsfunktionsanalyse von humanen Sprunggelenkdynamiken wie in Kapitel 2.5 beschrieben. Die verwendeten Daten sind die Ergebnisse eines stochastischen Inputs zwischen 1 und 50 Hz. Die hohen Werte der Kohärenz im oberen Diagramm der Abbildung sind Beweis für die Übereinstimmung des linearen Modells mit den experimentellen Daten. Der Gain-Plot weist dabei ein charakteristisches Verhalten auf und zwar einen konstanten Gain im Niederfrequenzbereich, ein Minimum bei mittlerer Frequenz und ein anschließender Anstieg der Kurve mit ca. 40 dB/Dekade bei hohen Frequenzen. Sichtbarer ist dieser charakteristische Gain-Verlauf in der folgenden Abbildung 2.9, hier ist der Gain in dB statt Nm/rad dargestellt. Die Darstellung in Dezibel erhält man durch die Multiplikation des Logarithmus zur Basis 10 des Wertes in Nm/rad mit dem Wert 20.

Die Art der Darstellung ist nicht einheitlich geregelt, weshalb in Abbildung 2.8 ein positiver Gain dargestellt ist und ein entsprechender Anstieg der Kurve im Hochfrequenzbereich, während Ab-

bildung 2.9 mit inversen Vorzeichen arbeitet, wodurch der Kurvenverlauf exakt gespiegelt ist. Selbiges gilt für die Kurvenverläufe der Phase, hier wäre im Falle der Abbildung 2.9 ein positives Vorzeichen der Phase zu erwarten. Beide Darstellungen im Bezug auf die Vorzeichen sind valide, im weiteren Verlauf der Thesis wird die Darstellung nach Abbildung 2.9 verwendet, also ein Gain vorwiegend im negativen Raum und Phasen im vorwiegend positiven Raum.

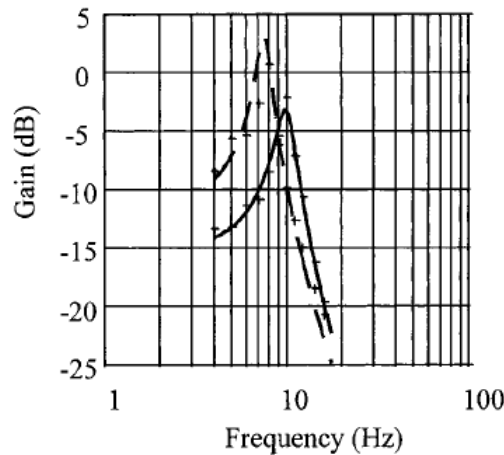


Abbildung 2.9: Beispielhafter Amplitudengang des muskuloartikulären Systems. Dargestellt ist einerseits eine 25 % MVC- (gestrichelt) und eine 50 % MVC-Kurve (durchgängig). Zu erwarten ist in beiden Fällen ein Anstieg der Kurve im Niederfrequenzbereich, ein Peak bei mittleren Frequenzen und ein drastischer Abfall der Kurve mit knapp -40 dB/Dekade bei hohen Frequenzen. Quelle: Lambertz et al. [13].

Die Phase in Abbildung 2.8 entwickelt sich von einem anfänglichen Wert von -180° bei niedriger Frequenz zu -90° bei dem Minimum bei mittlerer Frequenz, bis sie schließlich gegen 0° bei hohen Frequenzen geht.

Dieses Gesamtverhalten ist typisch für die Antwort eines Systems zweiter Ordnung, welches folglich zur Beschreibung und Modellierung von Gelenkdynamiken verwendet wurde. Das parametrische Modell hat die Form:

$$T(t) = I * \frac{d^2\Theta(t)}{dt^2} + B * \frac{d\Theta(t)}{dt} + K * \Theta(t) \quad (2.1)$$

Hierbei gilt: $T(t)$ = Drehmoment, $\Theta(t)$ = Winkelposition, I = Massenträgheitsmoment, B = viskoser Parameter, K = elastischer Parameter. Durch einen Vergleich mit einer nicht-parametrischen Variante dieser Funktion konnte eine Übereinstimmung der beiden von 97 % ermittelt werden, welches die Validität der Funktion 2.1 stützt [71].

Trotz der soliden Aussagekraft des parametrischen Modells sollte bedacht werden, dass die Abhängigkeit der Parameter von den Versuchsbedingungen wie Durchschnittsdrehmoment, Perturbationsamplitude oder Durchschnittsposition sehr hoch ist. Unter Berücksichtigung dieser Versuchsumgebung λ lässt sich ein erweitertes quasi-lineares Modell definieren. Dieses hat, auf-

bauend auf der Funktion aus Gleichung 2.1, folgende Form:

$$T(t) = I * (\lambda) * \frac{d^2\Theta(t)}{dt^2} + B * (\lambda) * \frac{d\Theta(t)}{dt} + K * (\lambda) * \Theta(t) \quad (2.2)$$

Die Auswirkungen von λ sind in Abbildung 2.9 sichtbar. Hier entspricht λ dem prozentualen MVC-Wert. Die zu erwartende Veränderung der Kurve durch Erhöhung des MVC ist eine Abnahme des anfänglichen Gains sowie ein niedrigerer und zu höheren Frequenzen verschobener Peak.

2.8 Modell zur Darstellung viskoelastischen Verhaltens

Aus dem Kapitel “Muskulatur” (Kapitel 2.2) ergeben sich die mechanischen Eigenschaften, welche in die Modellierung von Gelenkdynamiken einfließen. Aus den Erkenntnissen des Kapitels “Methoden zur Charakterisierung von Gelenkdynamik” (Kapitel 2.5) ergibt sich die mathematische Annäherung von Muskelverhalten, welche innerhalb der Kapitel “Quick Release” (Kapitel 2.6) und “Sinusoidal Perturbation” (Kapitel 2.7) in Messmethodiken verarbeitet ist. Außerdem sind im Rahmen des MARES die technisch-methodischen Vorgaben definiert. Um all diesen Anforderungen und Vorgaben gerecht zu werden, wird ein entsprechendes technisch umsetzbares Modell als Basis für die Konstruktion des Phantoms gewählt.

Die Muskelsteifigkeit beschreibt vereinfacht die Beziehung zwischen Verformung und Spannung die auf einen Körper wirkt. Es gibt verschiedene Rheologische Modelle die diese Beziehung für verschiedene Materialien beschreiben. Der Muskel kann vereinfacht als ein viskoelastisches Material beschrieben werden. Eines der Rheologischen Modelle ist das Kelvin-Voigt (KV)-Modell auf welches im Folgenden genauer eingegangen wird.

Die Charakterisierung im Sinne von passiven und aktiven Elementen der SEC findet sich im Rahmen des Kelvin-Voigt-Modells zur Simulation von Muskelverhalten wieder [72]. Diese Annäherung an echtes Muskelverhalten greift die passiven und aktiven Komponenten als federnde bzw. dämpfende Elemente auf [73]. Die resultierende viskoelastische Natur des Kelvin-Voigt-Modells spiegelt die Viskoelastizität von Muskeln wieder [74]. Ein viskoelastisches Modell wird unter anderem gewählt, da zahlreiche Eigenschaften viskoelastischer Materialien mit den Eigenschaften eines Muskels übereinstimmen: Hysterese im Spannungs-Dehnungs-Diagramm, Auftreten von Spannungsrelaxation (gleichbleibende Dehnung führt zu abfallender Spannung), Auftreten von Kriechverhalten (gleichbleibende Spannung führt zu zunehmender Dehnung). Das Kelvin-Voigt-Modell besteht aus einer Parallelschaltung einer Feder und eines Dämpfers, die Grundannahme ist hierbei, dass keine Biegungserscheinungen auftreten, die aufgebrachten Spannungen wirken also auf die Feder genauso wie auf den Dämpfer. Dieses Modell zur Simulation von Muskelverhalten dient als Grundlage für die Konstruktion des Phantoms, da es Muskelverhalten ausreichend genau simuliert und gleichzeitig eine berechenbare mechanisch-technische Konstruktion ermöglicht.

Kapitel 3

Methodik

In diesem Kapitel werden die Methoden zur Ermittlung der Ergebnisse erläutert, die zur Erfüllung des Ziels der Thesis vonnöten sind. Hierbei wird zunächst auf die für diese Arbeit angepasste Durchführung der Quick Release- und Sinusoidal Perturbation-Tests eingegangen. Darauf aufbauend, wird der Konstruktionsprozess des Phantoms beschrieben, in dem die Anforderungen an das Phantom und deren Umsetzung illustriert wird. Im Anschluss wird die Ermittlung der Inputvariablen dargestellt, welche benötigt werden, um das rechnerisch zu erwartende Verhalten des Phantoms zu prognostizieren. Dies umfasst die Bestimmung des Massenträgheitsmoments, der Federkonstanten sowie der Dämpfungskonstante. Anschließend werden diese Inputvariablen genutzt, um das angesprochene, rechnerisch zu erwartende Verhalten des Phantoms über ein Python-Skript zu modellieren. Dabei wird auch die Herleitung von Übertragungsfunktionen erläutert, welche für die Interpretation der SP-Tests benötigt werden. Es folgte eine Darstellung der in dieser Arbeit genutzten Testreihen sowie einer Erläuterung zur Datenverarbeitung und Weiterarbeitung der Rohdaten des MARES in der Theorie und in der Praxis mittels Python-Programmierung.

3.1 Quick Release

Das Quick Release Testverfahren (vgl. Abschnitt 2.6) zur Messung der SEC wird in der vorliegenden Arbeit auf das konstruierte Phantom angewendet. Es wird das QR-Verfahren nach Lambertz et al. verwendet [13].

Im Vergleich zum Quick Release an humanen Probanden, wird eine veränderte Konfiguration des MARES und der angebauten Arme genutzt (vgl. Abbildung 3.2 und Abbildung 3.5). Ein wesentlicher Unterschied zwischen der Testmethodik zum klassischen QR ist der Zeitpunkt, zu dem die Kraft und mit dem Hebelarm des MARES das Drehmoment auf die Achse des MARES wirkt. Für humane Probanden gibt das MARES per Befehl einen Zeitpunkt vor, zu dem der Proband Kraft auf die Fußplatte ausüben soll. Bei der Messung mit dem Phantom hingegen, liegt schon zuvor ein Moment an, sobald der SLA des MARES zu Beginn der Testreihe in Position fährt. Dies gilt sowohl für die Bestimmung des MVC als auch für die Durchführung der Testreihen. Ein weiterer Unterschied ist das unterschiedliche Trägheitsmoment des Phantoms und der gewöhnlichen Konfiguration mit Fußhalterung für die Probanden. Das Trägheitsmoment kann

zwar in der MARES Software für unterschiedliche Massen und damit Trägheitsmomente die an der Achse anliegen eingestellt werden, welche Implikation dies auf die Messungen hat, ist jedoch unklar und wird in Abschnitt 5.5.3 diskutiert. Bei humanen Probanden bewegt sich die Fußplatte bis zum Ende der Range of Motion. In den Phantom-Testreihen wurden die Endstops des MARES enger zueinander gesetzt, damit das MARES weniger Winkel während der Einstellung des MVC abfährt. Hierbei wird der Bewegungsspielraum der Auslenkung des SLA durch die Endstops begrenzt, da diese kurz vor Erreichen der Endstops einen Regelmechanismus aktivieren, der die Bewegung stoppt. Weiterhin ist der Bewegungsspielraum durch die Feder begrenzt, die die Bewegung stoppt, sobald sie nach Ausföjhrung des QR in ihre Ausgangslänge zurückgekehrt. Weiterhin werden bei Vermessung des Phantoms im Gegensatz zu humanen Probanden keine Pausen der Messung zur Entspannung der Muskeln durchgeführt.

3.2 Sinusoidal Perturbation

Zur Charakterisierung von Gelenkdynamiken während der Plantarflexion, wurden dem Sprunggelenk sinusoidale Perturbationen aufgeprägt. Hierbei wurde das Verfahren nach Lambertz et al. verwendet [13].

Grundsätzlich erfolgt die Testung wie in Kapitel 2.5, allerdings werden nun keine Humanprobanden, sondern das Phantom vermessen. Das Phantom ist so konzipiert, dass es auf möglichst einfache Art und Weise die gleiche Antwort des MARES induziert, wie sie auch bei der Testung mit Humanprobanden induziert würde. Hierbei ist die Methodik zur Erzeugung dieser Antwort vollkommen unterschiedlich, beeinflusst jedoch den Vorgaben entsprechend, die Antwort selbst nicht. Eine Besonderheit der SP-Tests mit dem Phantom ist, dass dauerhaft ein Drehmoment anliegt, auch während der Pausen die eigentlich dazu da sind, dem Probanden Zeit zum Entspannen zu geben. Auf Grundlage einiger Kontrolldurchläufe konnte festgestellt werden, dass dies zu keinen Problemen innerhalb des Systems föhrt. Da ein reproduzierbares Ergebnis zu erwarten ist, wurden alle Testreihen einmalig durchgeführt. Auf die Testreihen wird in Kapitel 3.3 genauer eingegangen.

Die resultierenden, frequenzabhängigen Veränderungen in Amplituden- und Phasengang spiegeln eine Mischung des Einflusses der mechanischen Elementen (Trägheitsmoment I , Elastizität K und Viskosität B) gemäß Gleichung 2.1 wider.

3.3 Das SPQR-Phantom

Zur Validierung der QR- und SP-Messmethode wird ein Phantom konstruiert und gebaut, dass die viskoelastischen Eigenschaften des *M.tricepsurae* abbilden soll. Das Phantom ersetzt in den durchzuföhrenden Testreihen den Probanden und es werden QR- und SP-Testläufe am Phantom durchgeführt. Durch die bekannten Parameter des Phantoms, soll eine Qualitätsbewertung der Messmethodik und des Messgerätes erfolgen. Da das Phantom für Quick Release sowie Sinusoi-

dal Perturbation Messungen entwickelt wird, trägt es den Namen SPQR-Phantom.

3.3.1 Lastenheft des Phantoms

Vor der Konstruktion des Phantoms wurde ein Lastenheft erstellt, welches eine Anforderungsliste an das Phantom beinhaltet. Diese Liste besteht aus Muss-Kriterien, die das Phantom notwendigerweise erfüllen muss, um der Aufgabe gerecht zu werden, sowie Soll-Kriterien, welche nicht umgesetzt werden müssen, jedoch wünschenswert wären. Alle Entwürfe des Phantomdesigns wurden unter Berücksichtigung der Anforderungsliste erstellt und auf Basis dieser evaluiert.

Muss-Kriterien:

- Vorhandensein einer viskosen und einer elastischen Komponente
Simulation der viskosen und elastischen Eigenschaften der Muskeln
Sollen parallel in einer Kelvin-Voigt Schaltung verbaut werden
- Möglichkeit verschiedene Steifigkeiten und Dämpfungen zu modellieren
Einstellung verschiedener Muskelparameter
Kann bspw. durch austauschbare Federn/Dämpfer oder Verstellbarkeit dieser realisiert werden
- Möglichkeit Drehmomente mit verschiedenen Abstufungen des MVC zu erzeugen
25, 50, 75 und 100 %
- Erzeugung realistischer Drehmomente
Drehmomente wie sie auch in vergangenen Studien (AGBRESA) erzeugt wurden müssen möglich sein (bis zu $\approx 200 \text{ Nm}$)
- Funktionalität bei QR und SP
Das Phantom muss hinsichtlich erzeugbarer Drehmomente und Stabilität für beide Versuchsreihen geeignet sein

Soll-Kriterien:

- Verstellbarkeit des MVC während des Versuches
Entweder Verstellbarkeit der Vorspannung unter Last oder Mechanismus zum temporären Lösen der Spannung
- Geringe eigene Trägheit des Phantoms oder gut abzuschätzendes Trägheitsmoment
Zur isolierten Betrachtung der Werte und Einstellung des Trägheitsmoments im System
- Möglichkeit verschiedene Trägheiten zu modellieren
Bspw. durch austauschbare Massen, Verschiebung der Massen entlang der Drehachse

3.3.2 Designprozess und Entwürfe

Aus den gegebenen Anforderungen wurden verschiedene Designentwürfe entwickelt, welche in Bezug auf ihre Praktikabilität und Umsetzbarkeit evaluiert wurden. Nachfolgend ist der Prozess bis zum finalen Designs des Phantoms dargestellt.

Abbildung 3.1a)-f) zeigt Skizzen der Entwürfe, welche zu Beginn des Designprozesses in Betracht gezogen und nach Evaluierung ihrer Qualität hinsichtlich des zu erreichenden Ziels verworfen wurden.

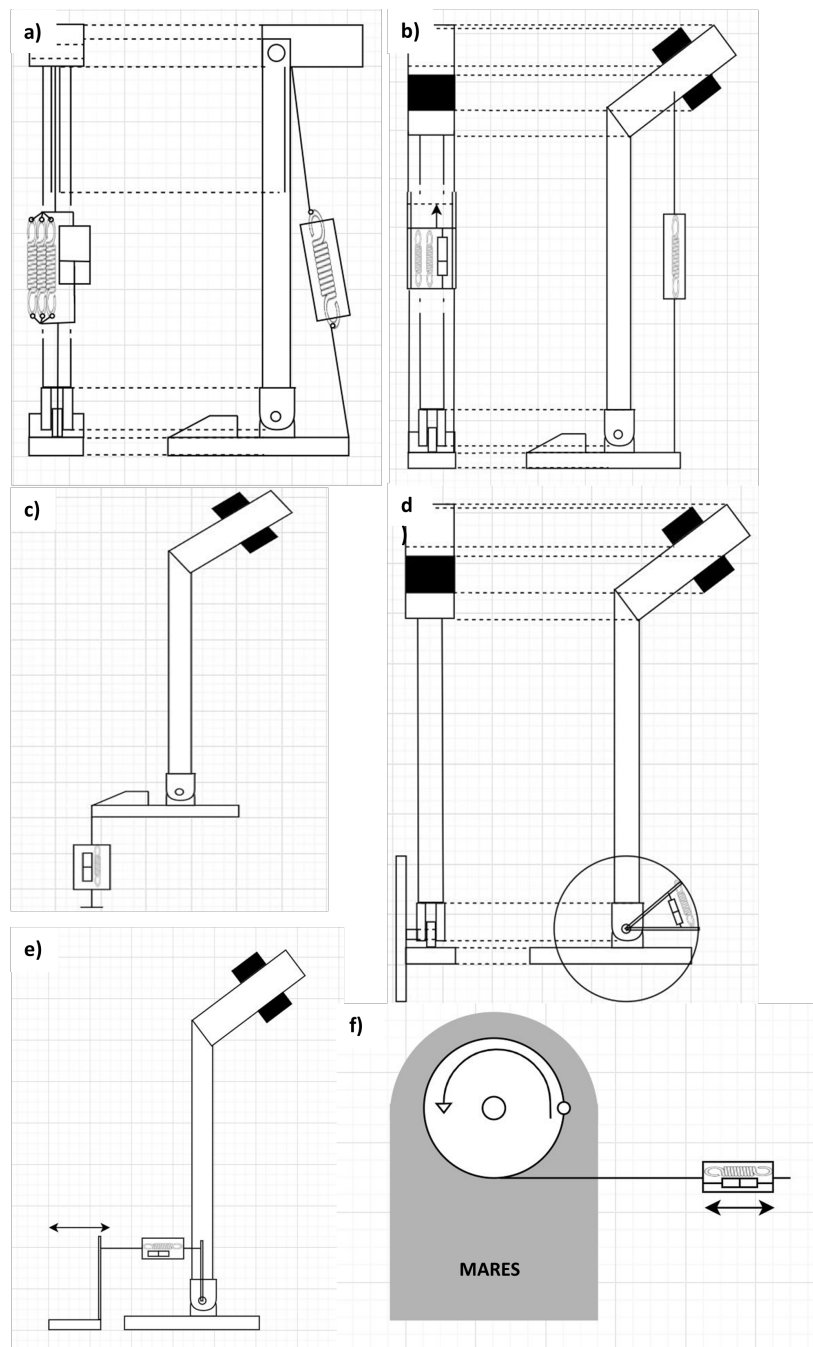


Abbildung 3.1: a)-f): Skizzen der verworfenen Designentwürfen des SPQR-Phantoms. Quelle: Eigene Abbildung.

Design 3.1a) zeigt den ersten Entwurf des SPQR Phantoms. Zu sehen ist ein vereinfachter Nachbau eines Beines mit Komponenten für Femur, Kniegelenk, die Tibia und Fibula zusammengefasst in einer Komponente, Sprunggelenk sowie Fuß. Der KV-Komplex als viskoelastische Komponente wird von der Ferse zu einer Schiene geführt, die dorsal am Unterschenkel angebracht ist. Die Spannung des KV-Komplexes erfolgt über Verschiebung der Befestigung auf der

Schiene. Federn und Dämpfer können in beliebigen Kombinationen ein- und ausgehen werden, um verschiedene Federsteifigkeiten und Dämpfungseigenschaften zu simulieren.

Durch die nicht orthogonale Anbringung der Zugvorrichtung an der Ferse und den so entstehenden Winkel besitzt dieses Phantom keine lineare Kraftübertragung was Berechnungen im späteren Verlauf unnötig erschwert. Zudem ist das Kniegelenk für vorliegende Versuche redundant, da sich das Phantom in dieser Achse weder bei Quick Release noch bei Sinusoidal Perturbation Versuchen bewegt. Weiterhin wird der Dämpfer in diesem Versuchsaufbau auf Druck belastet, was komplett steife Komponenten zwischen beiden Befestigungen des Dämpfers voraussetzt. Dies ist jedoch nicht mit einer Verschiebung auf einer Schiene und der daraus resultierenden Längenänderung des KV-Komplexes vereinbar. Aus genannten Gründen wurde dieses Phantom früh im Designprozess verworfen.

Das Design 3.1b) zeigt eine weiterentwickelte Form von Design 3.1a). Das Kniegelenk wurde entfernt und Feder und Dämpfer ziehen die Ferse senkrecht nach oben zum Oberschenkel. Der Auszug der Feder ließe sich in diesem Design nicht mehr durch Längenänderung der kompletten Zugvorrichtung durchführen. Daher wurde ein Rahmen konzipiert, der Federn und Dämpfer ausziehen kann, ohne die Gesamtlänge zu verändern. Dies würde über ein Raster passieren, in das man den KV-Komplex einrasten könnte.

Dieses Design wurde verworfen, da aufgrund der Einspannung des Phantoms am Oberschenkel die Zugvorrichtung sehr nah am Unterschenkel des Modells entlanggeführt werden müsste. Dies würde zu einem kürzeren Hebelarm und damit zu einer zu schlechten Kraftübertragung führen, um die gewünschten Momente zu erreichen.

Design 3.1c) sieht eine plantare Befestigung des KV-Komplexes vor. Hier könnten deutlich höhere Drehmomente erreicht werden, das Design wurde jedoch verworfen, da die plantare Befestigung das Einspannen des Phantoms ins MARES erschwert.

Design 3.1d) zeigt einen Entwurf, bei dem ein Stab fest mit der Achse des Sprunggelenkes verbunden ist und sich bei Bewegung des Fußes auf einer Kreisbahn bewegt. Zwischen diesem Stab und einem zweiten Stab ist der KV-Komplex eingefasst. Der zweite Stab lässt sich auf der kreisförmigen Bahn verschieben und einrasten und kann so ein Drehmoment erzeugen, das die gewünschte Bewegung in Plantarrichtung hervorrufen würde. Wie in Design 3a) ist die Berechnung der Kräfte in diesem Modell nur unter Berücksichtigung des Winkels zwischen beiden Stäben möglich.

Design 3.1e) umgeht das Problem des Winkels, indem Feder und Dämpfer vom Stab, der mit der Achse verbunden ist, senkrecht weggeführt werden.

An dieser Stelle im Designprozess wurde die Entscheidung getroffen, kein komplettes Modell des Beines zu konzipieren, sondern ein Phantom, welches lediglich auf die viskoelastische Komponente des M. triceps surae reduziert ist. Der so simplifizierte Aufbau soll Messunsicherheiten verhindern und mögliche Einflussfaktoren auf die Steifigkeit, wie ein erhöhtes Trägheitsmoment, minimieren um die Steifigkeit isoliert zu betrachten.

Vor diesem Hintergrund wurde Design 3.1f) entwickelt, bei dem ein kreisförmiger Aufsatz direkt

an der Drehachse des MARES befestigt ist. Der SLA sowie die Einfassung des Fußes für den Quick Release mit Probanden sind in diesem Aufbau entfernt. Auf dieser Kreisbahn befindet sich eine Führung in der ein Stahlseil liegt und die komplette Scheibe in Drehung versetzt und damit an der Achse des MARES ein Drehmoment erzeugt.

3.3.3 Finales Design des SPQR-Phantoms

Das finale Design des Phantoms basiert auf Design 3.1f. Statt eine Scheibe neu zu konstruieren, werden viele Komponenten genutzt, die zum Standard-Equipment des MARES gehören und somit bereits vorhanden sind.

Bei QR- und SP-Tests ist an der Achse des MARES der SLA befestigt, an welchem gewöhnlich die Einfassung für den Fuß des Probanden angebracht ist (vgl. Abbildung 2.5). Diese Fußhalterung ist für diesen Aufbau zur Reduzierung der Trägheit abgebaut (Abbildung 3.2a). Weiterhin sind die am static lever befestigten Kissen zur Einfassung des Oberschenkels entfernt, um zusätzliche Störgrößen zu verhindern.

Der static lever wird waagerecht ausgerichtet. Am static lever wird senkrecht nach unten der medium lever angebracht, ein weiterer Arm, der zum Equipment des MARES gehört (Abbildung 3.2b).

Auf short- und medium lever werden Adapter (Fixing Adapter) geschoben, die mit ihren zylinderförmigen Enden als Übergang zu anderen Bauteilen des MARES dienen (Abbildung 3.2a).

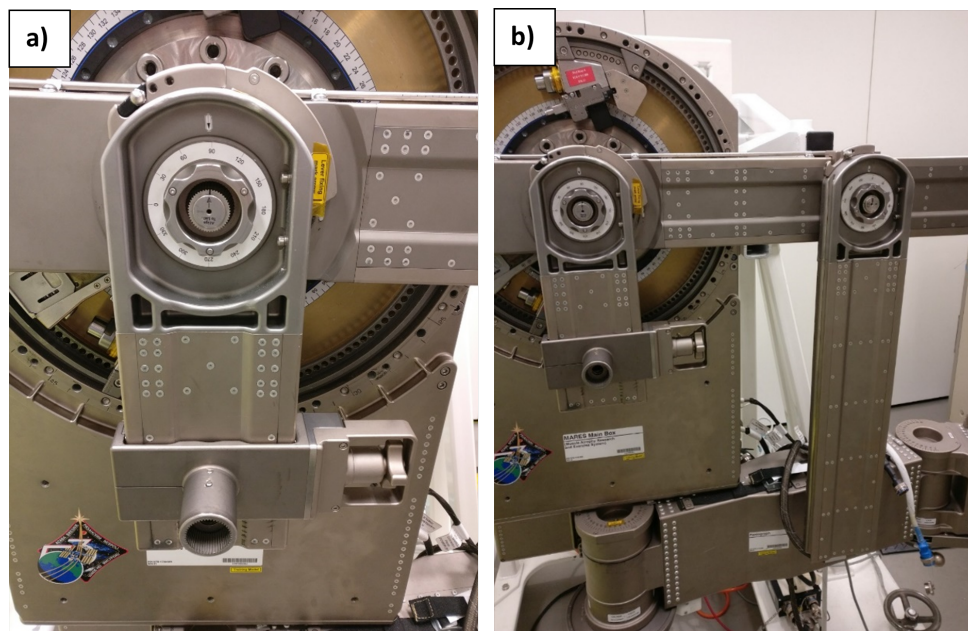


Abbildung 3.2: a) SLA des MARES mit Fixing-Adapter der auf die Arme des MARES aufgesteckt und verschoben werden kann. b) Das MARES mit befestigtem SLA (kürzerer Arm) und medium lever (längerer Arm) am static lever (horizontaler Arm). Quelle: eigene Fotografie.

Es wurden zwei zylinderförmige Aufsätze konstruiert (Abbildung 3.3), die auf die Adapter am SLA und am medium lever des MARES aufgesteckt und mit diesen verschraubt werden können. Die Schraube wird dabei von vorne in den Hohlraum eingeführt und mit einer Mutter auf der anderen Seite befestigt (Abbildung 3.4).

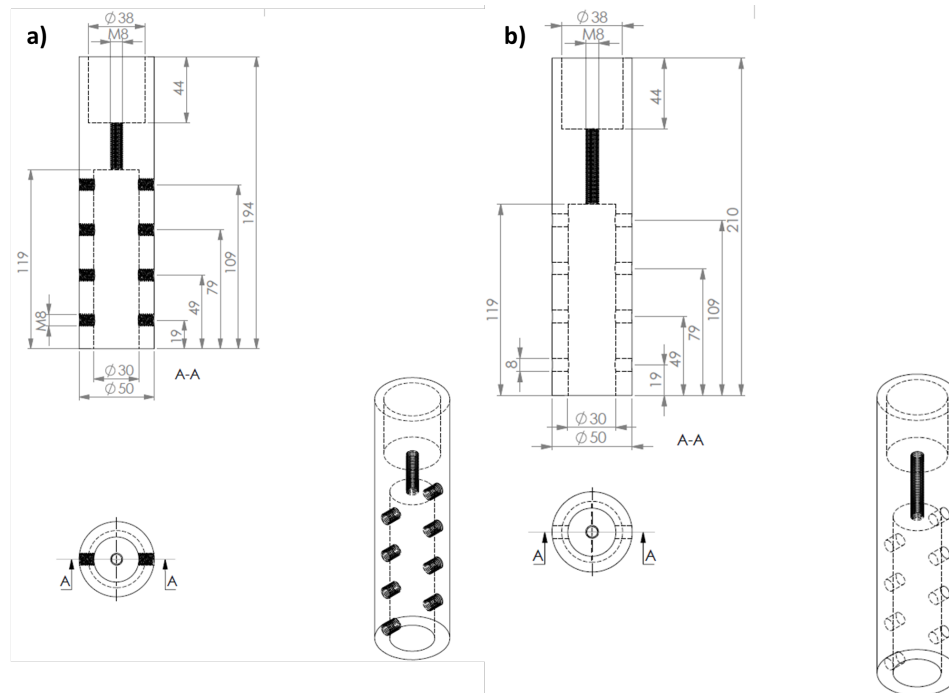


Abbildung 3.3: Konstruktionszeichnung der SPQR-Aufsätze für das MARES mit: a) 194 mm langer Aufsatz mit M8 Gewindebohrungen und b) 210 mm langer Aufsatz mit 8mm Bohrungen ohne Gewinde. Quelle: Eigene Abbildung.

Die Fertigung der Aufsätze erfolgte durch das Unternehmen Schmitz in Eitorf. Um eine ausreichende Festigkeit zu gewährleisten wurden die Aufsätze aus Stahl gefertigt. Die fertigen Aufsätze sind in Abbildung 3.4 zu sehen.

In die Bohrungen der Aufsätze können Augenschrauben eingehängt und mit Muttern auf der Gegenseite befestigt werden. Die Federn werden mit Schäkeln in die Augenschrauben eingehängt und sind während der QR und SP-Messungen durchgehend gespannt. Der Dämpfer wird mit den Augenschrauben fest verschraubt, sodass er steif befestigt ist und damit auch eine Druckbelastung dämpfen kann.

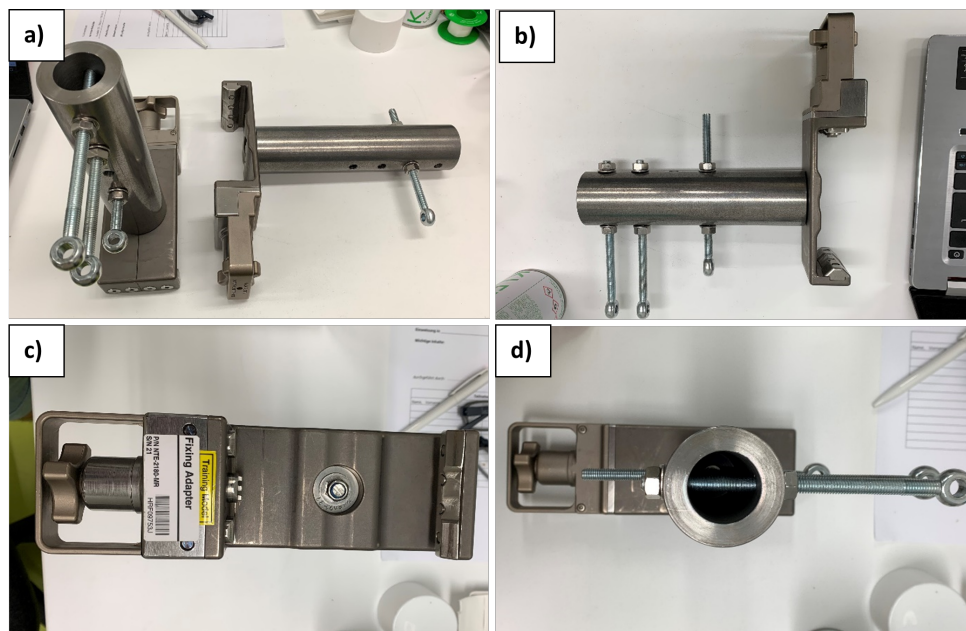


Abbildung 3.4: a) Beide Stahlaufsätze mit zugehörigen Fixing-Adaptern und Augenschrauben zur Befestigung der Federn und Dämpfer. b) rechter Aufsatz in der Seitenansicht. c) Rückansicht der Befestigung des Aufsatzes auf dem Fixing Adapter. d) Vorderansicht des Aufsatzes. Quelle: Eigene Abbildung.

Der eine Aufsatz ist mit 194 mm kürzer als der zweite Aufsatz, welcher eine Länge von 210 mm aufweist. Die zusätzliche Länge des einen Aufsatzes von 16 mm ist vonnöten, um eine Unebenheit der beiden Halterungen für die Aufsätze am MARES auszugleichen. Diese sind um 16 mm zueinander verschoben.

Außerdem unterscheiden sich beide Aufsätze durch die seitlich befindlichen Bohrungen. Der kürzere der beiden Aufsätze hat Gewindebohrungen, in die die Augenschrauben direkt eingeschraubt werden. Der längere Aufsatz hat normale Bohrungen ohne Gewinde. Die Augenschrauben werden hier mit einer Mutter an der gegenüberliegenden Seite des Aufsatzes gehalten.

Durch diesen Aufbau können an einem Aufsatz die Augenschrauben auf verschiedene Federlängen eingestellt werden, am anderen Aufsatz dann während des Versuches die Federn aufgezo- gen werden, um unterschiedlich große Kräfte zu erzeugen. Die Verwendung einer Seite ohne Gewindebohrungen und einer Mutter als Befestigung bewirkt dabei, dass die Augenschrauben sich beim Drehen der Mutter gerade durch die Bohrung bewegen. Dies verhindert, dass die Federn und Dämpfer beim Aufziehen Torsionskräften ausgesetzt sind.

Ist das Phantom komplett aufgebaut (Abbildung 3.5a) so sind Federn und Dämpfer zwischen beiden Aufsätzen parallel verbaut und bilden somit einen KV-Körper (vgl. Abschnitt 2.8). Die beiden Aufsätze zusammen mit dazwischen befestigten Federn und Dämpfern bildet das gesamte SPQR-Phantom.

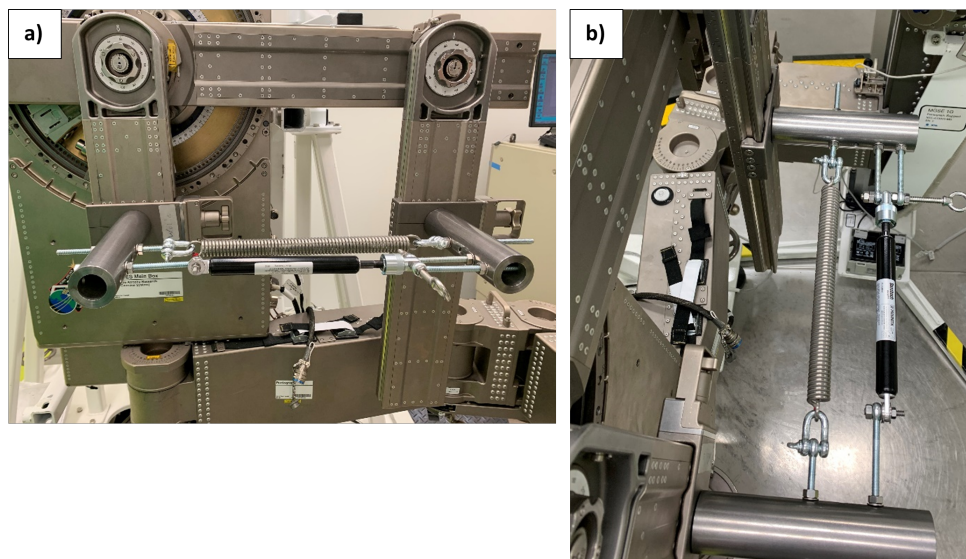


Abbildung 3.5: a) Seitliche Ansicht des MARES in der genutzten Konfiguration mit vollständig eingebautem SPQR-Phantom. b) Ansicht des SPQR-Phantoms von oben. Quelle: eigene Fotografie.

Die Federn spannen erst, wenn das MARES zu Beginn des Quick Release Versuches in die Ausgangsposition, die sogenannte Nullstellung, fährt. Die Nullstellung wurde so konfiguriert, dass der SLA senkrecht zum static lever nach unten ausgerichtet ist. Im gespannten Zustand befinden sich short und medium lever in einer Parallelstellung (Abbildung 3.2), sodass die Feder senkrecht zum SLA steht und in einem 90° Winkel an diesem zieht. Das erzeugte Drehmoment hängt somit direkt über

$$M = F * r \quad (3.1)$$

von der Federkraft und dem festgelegten Abstand des linken Phantomaufsatzes zur Drehachse des MARES ab. Dieses Drehmoment simuliert also das Drehmoment, welches klassisch im QR-Versuch vom Probanden durch eine Plantarflexion auf das Gerät gegeben wird. Statt also auf die Fußplatte zu drücken, wird die Achse in die entgegengesetzte Richtung gezogen. Dies verhindert Querkräfte und Biegungen der Feder, die potenziell bei einem System mit einer Druckfeder auftreten könnten. Dabei ist der Auszug der Federn/Dämpfer über die ein- und ausschraubbaren Gewindestangen regelbar und somit die Kraft mit der gezogen wird variierbar. Die Federkonstanten und Dämpfeigenschaften sind zusätzlich flexibel, da sowohl Federn als auch Dämpfer ausgetauscht werden können.

3.4 Parameterabschätzung und -validierung

Die Parameter der Gleichung 2.1 lassen sich auf unser Modell übertragen. Hierbei gilt: I ist das Massenträgheitsmoment des Gesamtsystems, B die rotatorische Dämpfungskonstante des verwendeten Dämpfers und K ist die Drehfederkonstante der genutzten Feder. Die Methodik zur

Bestimmung des Wertes der Parameter wird im Folgenden beschrieben.

3.4.1 Massenträgheitsmoment I

Das Massenträgheitsmoment wurde experimentell bestimmt. Hierbei wurde die Methodik zur praktischen Bestimmung von Trägheitsmomenten starrer Körper mit Hilfe von Drehschwingungen genutzt [75]. Der zu untersuchende Körper ist das gesamte System, dessen Massenträgheit maßgeblich in die Messung einfließt. Das heißt in unserem Falle der SLA, welcher an der Drehachse angebracht wird, sowie der Adapter, welcher in einer gewissen Distanz an dem SLA angebracht wird, der wiederum die Aufnahme für unser Modell bildet. Das Massenträgheitsmoment des Gesamtsystems wird nun über das Drehschwingungsverfahren gemessen. Wie in der oben angegebenen Literatur beschrieben, wird hierfür eine möglichst reibungsarme Achse verwendet, die mit einem Oszillator verbunden ist. Anschließend wird der zu untersuchende Körper auf der Achse befestigt und der Oszillator durch Drehung um den Winkel j gespannt. So entsteht das rücktreibende Drehmoment $M = -D_{RM}j$, hierbei ist D_{RM} das Richtmoment des Oszillators. Die Bewegungsgleichung unter Vernachlässigung der Reibung ergibt sich zu:

$$I_0 * j'' = -D_{RM} * j \quad (3.2)$$

I_0 ist hierbei das Trägheitsmoment der Apparatur. Die Linearität des Richtmomentes D_{RM} , kann hier angenommen werden, da nur geringe Auslenkwinkeln j vorliegen. Gilt die Anfangsbedingung $j(0) = 0$, so folgt für die obige Gleichung:

$$\varphi = a \sin\left(\sqrt{\frac{D_{RM}}{I_0}} * t\right) \quad (3.3)$$

Die Apparatur führt also eine harmonische Schwingung aus, dessen Schwingungsdauer T_0 durch nachstehende Formel beschrieben werden kann:

$$T_0 = 2\pi\left(\sqrt{\frac{I_0}{D_{RM}}}\right) \quad (3.4)$$

Anders als in der Literatur angegeben, wird nun eine rechteckige Platte bekannter Abmaße konzentrisch auf die Drehachse der Apparatur "gesteckt" (Abbildung 3.6. Zur Bestimmung des Trägheitsmomentes des Aufbaus I_0 , wird nun ein Referenzkörper bekannten Durchmessers d und Masse m auf der Platte befestigt und vermessen. Das Trägheitsmoment erhöht sich, nach Befestigung des zylinderförmigen Referenzkörpers, auf $I = I_0 + I_R$, wobei I_R mathematisch bestimmbar ist und sich wie folgt berechnet:

$$I_R = \frac{1}{2} * m_R * \left(\frac{d}{2}\right)^2 \quad (3.5)$$

Entsprechend ergibt sich die Schwingungsdauer des Referenzsystems T_R zu:

$$T_R = 2\pi \left(\sqrt{\frac{I_0 + I_R}{D_{RM}}} \right) \quad (3.6)$$

Zur Messung der beiden Schwingungsdauern T_0 und T_R wurde der Tisch, auf welchem unser Gesamtsystem fixiert wurde, mit Hilfe des Oszillators, in unserem Fall ein Theraband, in Schwingung gebracht, der Vorgang per Smartphone-Kamera bei einer Bildwiederholfrequenz von 60 Hz dokumentiert und anschließend über die Software Kinovea die Schwingungsdauer eines Markerpunktes auf dem Tisch analysiert. Mit Hilfe dieser Werte lässt sich D_{RM} wie folgt bestimmen:

$$D_{RM} = \frac{(2\pi)^2 * I_R}{T_R^2 - T_0^2} \quad (3.7)$$

Aus der Formel 3.4 folgt für I_0 :

$$I_0 = \frac{T_0^2 * D_{RM}}{4 * \pi^2} \quad (3.8)$$

Bei bekanntem I_0 lässt sich nun das Trägheitsmoment I eines beliebigen Körpers ermitteln.

$$I = \frac{T^2 * D_{RM}}{4\pi^2} - I_0 \quad (3.9)$$

Sofern die Drehachse des Untersuchungsobjektes nicht mit dem Massenschwerpunkt übereinstimmt, muss mit Hilfe des Satzes von Steiner die Entfernung zwischen Schwerpunkt und Drehachsenmittelpunkt berücksichtigt werden. Bei bekanntem Gewicht des Untersuchungsobjektes, sowie des Abstandes seiner Drehachse von seinem Massenschwerpunkt, kann nun die nachstehende Formel zur Berechnung des angepassten Trägheitsmoments genutzt werden:

$$I_v = I + m * d^2 \quad (3.10)$$

Nachfolgend ist der Versuchsaufbau zur Bestimmung des Massenträgheitsmoments des Phantoms dargestellt.



Abbildung 3.6: Messaufbau zur Bestimmung des Massenträgheitsmoments des Phantoms. In den Schraubstock eingespannt ist die nahezu reibungsfrei gelagerte Platte, auf welcher sich wiederum zentral der dunkelgraue Referenzkörper befindet. In den Ecken der Platte sind Markerkreuze angebracht, die zur späteren Auswertung verwendet werden. Das blaue Theraband wirkt als Oszillator. Zu Testbeginn wird die Platte händisch ausgelenkt und der Prozess per Smartphonekamera aufgenommen. Quelle: Eigene Fotografie.

3.4.2 Rotatorische Dämpfungskonstante B

Die rotatorische Dämpfungskonstante B eines Dämpfers wird in zwei Schritten ermittelt. Zunächst muss die lineare Dämpfungskonstante gemessen und anschließend in eine rotatorische Form übersetzt werden. Die lineare Dämpfungskonstante kann durch das Anhängen einer definierten Masse m an den in aufrechter Position befindlichen Dämpfer ermittelt werden. Anschließend wird die Zeit t gemessen, innerhalb derer die Masse den Dämpferhub l durchquert. Nach Vorbereitung der Variablen, kann hieraus zunächst die lineare Dämpfungskonstante und anschließend die rotatorische Dämpfungskonstante abgeleitet werden. Die vorzubereitenden Variablen sind die Dämpfungskraft F und -geschwindigkeit v . Dies geschieht nach folgender Formel:

$$F = m * g, v = \frac{l}{t} \quad (3.11)$$

sowie daraus ableitbar das Dämpfungsrehmoment M und die Winkelgeschwindigkeit ω bei bekanntem Radius r :

$$M = F * r, \omega = \frac{v}{r} \quad (3.12)$$

Schließlich ergibt sich für die rotatorische Dämpfungskonstante B:

$$B = \frac{M}{\omega} \quad (3.13)$$

3.4.3 Drehfederkonstanten K

Zur Untersuchung des Einflusses der Drehfederkonstante K auf die vorgeschlagene Modellierung gemäß 2.1, werden insgesamt vier unterschiedliche Federn verwendet, deren translatorische Federkonstanten in einem späteren Schritt in rotatorische Konstanten umgerechnet werden. Die verwendeten Federn haben laut Herstellerangaben lineare Federkonstanten ($R_1 = 1,48 \text{ N/mm}$, $R_2 = 2,89 \text{ N/mm}$, $R_3 = 4,553 \text{ N/mm}$, $R_4 = 8,763 \text{ N/mm}$). Die Linearität der Federkonstanten wird durch eine kurze Testreihe verifiziert, in der die Federn unterschiedlich stark ausgelenkt werden (Δl) und die entsprechend anliegende Kraft (F) gemessen wird. Der Versuchsaufbau besteht dabei aus einer eingespannten Feder in horizontaler Position, die unter Nutzung einer Kofferwaage ausgelenkt wird, wodurch die Kraft erfasst werden kann. Die Auslenkung erfolgt horizontal und wird mit Hilfe eines Maßbandes gemessen. Anschließend werden die Messpunkte aufgetragen und die Geradengleichung ermittelt. Der Wert der Steigung s entspricht der Federkonstanten. Mathematisch bedeutet dies:

$$s = \frac{y}{x} \rightarrow s = \frac{F}{\Delta l} \quad (3.14)$$

Die ermittelten Federkonstanten sind translatorische Kenngrößen und können anschließend in eine rotatorische Form (K) überführt werden. Dies geschieht nach folgender Formel:

$$U = 2\pi * r, \rho = \frac{U}{360^\circ} * \frac{180^\circ}{\pi}, K = \rho * R * r \quad (3.15)$$

Hierbei gilt: U = Umfang, ρ = 1rad von U .

3.5 Testreihen

In diesem Kapitel wird die Struktur und Durchführung der Testreihen zur Vermessung des Phantom-Modells beschrieben. Bei allen Testreihen wird über die Funktion „Angle Torque Relation“ der MARES-Software die für die Messung gewünschte MVC eingestellt. Die Einstellung der MVC erfolgt im Falle des Phantoms händisch indem gegen den SLA gedrückt wird, bis die entsprechende MVC erreicht wird. Das Drehmoment wird währenddessen am Bildschirm des MARES-Laptops abgelesen.

Die Quick Release Tests wurden bei einer MVC von 40 Nm sowie 80 Nm durchgeführt. Sinusoidal Perturbation erfolgte bei einer MVC von 80 Nm und 120 Nm.

Alle Federn werden mit und ohne Dämpfer für sowohl QR als auch SP vermessen.

3.5.1 Quick Release

Entsprechend der Gesamtheit aller Kombinationen der veränderbaren Parameter werden die Testreihen der QR Tests in Tabelle 3.1 erfasst.

Tabelle 3.1: Testreihen der QR-Untersuchungen. Dargestellt sind alle untersuchten Kombinationen aus Dämpfungskonstante B, maximaler Muskelkontraktion MVC, den verschiedenen Federn mit ihren Drehfederkonstanten K, die genutzten Abstufungen des MVC %MVC sowie das voreingestellte Trägheitsmoment I.

MVC [Nm]	K [Nm/rad]	B [Nms/rad]	%MVC [%]	I [kgm ²]
40	80	0	25/50/75	0,35
40	158	0	25/50/75	0,35
40	248	0	25/50/75	0,35
40	478	0	25/50/75	0,35
80	158	0	25/50/75	0,35
80	248	0	25/50/75	0,35
80	248	0	25/50/75	0,097
80	478	0	25/50/75	0,35
40	80	105	25/50/75	0,35
40	158	105	25/50/75	0,35
40	248	105	25/50/75	0,35
40	478	105	25/50/75	0,35
80	158	105	25/50/75	0,35
80	248	105	25/50/75	0,35
80	248	105	25/50/75	0,097
80	478	105	25/50/75	0,35

Die Feder K_{80} wurde nicht für eine MVC von 80 Nm vermessen, da die Feder die geforderten Drehmomente durch eine zu niedrige Federkonstante nicht aufbringen kann.

Das standardmäßig eingestellte Trägheitsmoment bei allen Messungen beträgt 0,35 kgm². Die QR-Messungen für K_{248} wurde mit einem voreingestellten Trägheitsmoment von 0,097 kgm² wiederholt, um den Einfluss des eingestellten Trägheitsmoments auf die Messergebnisse zu überprüfen.

Wie in Abschnitt 2.6 beschrieben erfolgen für jedes %MVC drei einzelne QR Messungen. Für jede Kombination aus MVC, K und B werden also neun einzelne QRs in drei Abstufungen des maximalen Drehmoments durchgeführt.

Anhand von Tabelle 3.1 ergeben sich so 144 einzelne Quick Releases, die im Verlauf der Messungen durchgeführt werden.

Tabelle 3.2: Exemplarische Darstellung des Messprotokolls für ausgewählte QR-Testläufe.

Testreihe	Dateiname	Datum	Uhrzeit	Feder	MVC [Nm]	Reihenfolge %MVC	Dämpfer	Kommentar
ATR	06Oktober2020 [10h55m04s]	06. Okt	12:38	/	80	/	nein	Festlegung des MVC
QR	06Oktober2020 [10h55m04s]	06. Okt	12:45	K_{248}	80	75, 50, 25	nein	/
QR	06Oktober2020 [10h55m04s]	06. Okt	12:56	K_{478}	80	50, 25, 75	nein	1. QR: zu hohes Drehmoment
QR	06Oktober2020 [10h55m04s]	06. Okt	13:10	K_{158}	80	75, 25, 50	nein	/

Tabelle 3.2 zeigt exemplarisch, wie während der Messung der Versuchsaufbau dokumentiert wurde. Durch die während der Messung dokumentierte Uhrzeit sowie der dokumentierten Reihenfolge der %MVC, können die einzelnen Quick Releases auf den vom MARES ausgegebenen Dateien den Parametern zugeordnet werden.

Komplikationen, die während der Messung aufgetreten sind, wurden in der Spalte „Kommentar“ festgehalten. Beispiele hierfür sind ein Nicht-Erreichen des geforderten Drehmoments oder das Lockern von Schrauben des Phantoms während der Testreihe.

Alle QR-Messungen wurden mit den voreingestellten Endstoppositionen für QR-Messungen durchgeführt, wie sie auch bei AGBRESA genutzt wurden. Alle Messungen in ihrer Gesamtheit wurden im späteren Verlauf der Arbeit mit angepassten weiter auseinander liegenden Endstoppositionen wiederholt, nachdem die eigentlichen Endstoppositionen für die Quick Release Messung am Phantom als nicht geeignet eingestuft wurden. Die Endstops lagen für die Phantommessung zu nah beieinander und ein Gegenregelmechanismus des MARES nahm Einfluss auf die Messungen. Die in Abschnitt 4.2 dargestellten Ergebnisse beziehen sich ausschließlich auf die geänderten Positionen der Endstops. Eine nähere Erläuterung findet sich im Abschnitt 5.5.

3.5.2 SP-Testreihen

Im Falle der SP-Messreihen werden, entsprechend der Gesamtheit aller Permutationen, die Messdaten pro Frequenz entsprechend Tabelle 3.3 erfasst. Hierbei werden die Federn K_{158} , K_{248} und K_{478} verwendet. Die Feder mit der geringsten Federkonstante (K_{80} , 80 Nm/rad) wird nicht verwendet, da die Auslenkung zur Abdeckung aller %MVC-Wert im Rahmen der SP-Testreihen nicht erreicht werden konnte.

Tabelle 3.3: Auszug der Testreihen der SP-Untersuchungen. Dargestellt sind ein Teil der untersuchten Permutationen aus Dämpfungskonstante B, maximaler Muskelkontraktion MVC, Drehfederkonstanten K sowie Muskelaktivierung %MVC. Für jede dieser Permutationen werden die Daten bei Frequenzen 4 Hz bis 16 Hz erfasst. Nicht dargestellt sind die Testreihen für $B = 105 \text{ Nms/rad}$, für $MVC = 120 \text{ Nm}$ und den sich daraus ergebenden weiteren Permutationen.

MVC	B	K	%MVC	4 Hz	5 Hz	6 Hz	7 Hz	8 Hz	9 Hz	10 Hz	12 Hz	14 Hz	16 Hz
80	0	158	0										
80	0	158	0,25										
80	0	158	0,5										
80	0	158	0,75										
80	0	248	0										
80	0	248	0,25										
80	0	248	0,5										
80	0	248	0,75										
80	0	478	0										
80	0	478	0,25										
80	0	478	0,5										
80	0	478	0,75										

3.6 Datenverarbeitung

In diesem Kapitel wird die grundsätzliche Datenverarbeitung und -aufbereitung erläutert. In entsprechenden Unterkapiteln wird die spezielle Datenverarbeitung für die QR- bzw. SP Messmethode beschrieben.

Die QR- und SP-Rohdaten, welche sich nach Abschluss der Testdurchführungen auf dem MARES befinden, werden auf den MARES-Laptop übertragen und die kodierten Daten mittels einer LABView- Funktion dekodiert. Durch die Dekodierung werden die Rohdaten zu auslesbaren Streams die nach den verschiedenen gemessenen Eigenschaften getrennt sind. Die für die QR und SP-Auswertung relevanten Streams sind dabei:

1. „Subject_Torque“: Verlauf des Drehmoments an der Drehachse über das gemessene Zeitintervall
2. „Angular_Position“: Verlauf der Winkelposition an der Drehachse und damit des SLA über das gemessene Zeitintervall
3. „Angular_Velocity“: Verlauf der Winkelgeschwindigkeit über das gemessene Zeitintervall
4. „Time_Mark_1“ und „Time_Mark_2“: Automatisch gesetzte Zeitstempel die am Start und Ende jedes einzelnen Quick Releases sowie jeder Sinusoidal Perturbation gesetzt und als lesbare Datenstreams ausgegeben werden. Dadurch ist jeder einzelne QR und SP mit einer Zeitmarkierung versehen

In Abbildung 3.7 ist exemplarisch ein Positionsstream des SLA sowie ein Stream des Drehmomentes eines in dieser Arbeit genutzten Datensatzes dargestellt. Die Darstellung wurde mit Hilfe des R-Skriptes *proc_MARES_mv_v0.2* erstellt. Dieses Skript ist ein eigens programmiertes R-Skript zur Auswertung und Darstellung der QR-Daten, wurde aber im späteren Verlauf durch das Python Skript *mares_QRanalysis4 - 0.py* ersetzt (vgl. Abschnitt 3.6.2). Der zugehörige Programmcode befindet sich im Anhang (vgl. Anhang A.3).

Diese dekodierten Daten bilden die Grundlage für die separate Weiterverarbeitung im Rahmen der QR- bzw. SP-Auswertung.

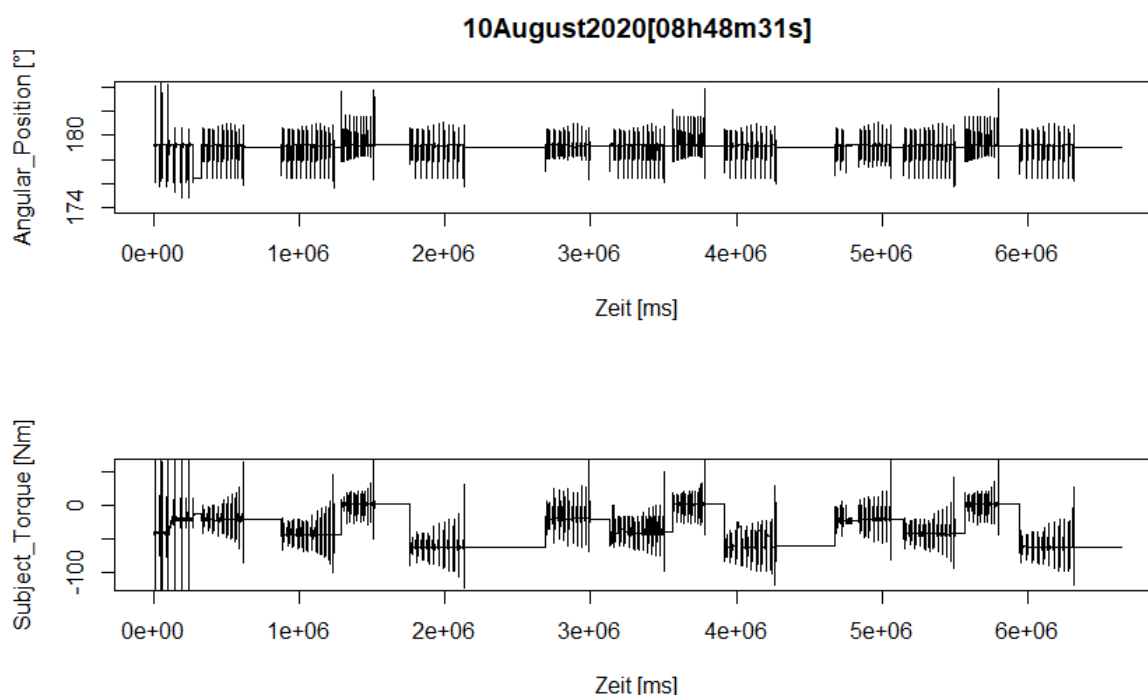


Abbildung 3.7: Darstellung des Streams der SLA-Position *Angular_Position* (oben) sowie des Drehmomentes *Subject_Torque* (unten) über der gemessenen Zeit seit Start des MARES. Zu sehen sind die Datenstreams einer SP-Messreihe. Quelle: Eigene Abbildung.

3.6.1 Grundlagen der Auswertungsmethodik des Quick Release

Die durch die Versuche am MARES erhaltenen Daten werden grundlegend ausgewertet wie von Lambertz et al. in vergangenen Studien zur Atrophie von Muskeln nach Langzeit-Raummissionen [13]. Zur Bestimmung der Muskelsteifigkeit werden die ersten Millisekunden (in vorliegender Arbeit 15 ms) der Bewegung nach Auslösen des Quick Release betrachtet. *in vivo* liefert diese Methodik die Möglichkeit die Messung lediglich auf die Series Elastic Components der Muskeln zu beziehen [26][29][76]. In dieser Zeit haben noch keine Reflexe eingesetzt und der Muskel kann isoliert betrachtet werden [77].

Für jeden QR Testlauf werden folgende Parameter analysiert:

1. Das isometrische Drehmoment (T_{iso}) kurz vor Auslösen des Quick Release
2. Die Änderung der Winkelposition ($\Delta\Theta$)
3. Die Änderung der Winkelbeschleunigung ($\Delta\Theta''$, als zweite Ableitung von $\Delta\Theta$) 15 ms nach Erreichen der maximalen Beschleunigung nach Auslösen des Quick Release
4. Das Trägheitsmoment (I)

Die Trägheit wird dabei als konstant angenommen. Dies kann durch Betrachtung des Übergangs von statischer Phase und dynamischer Phase überprüft werden. Zu diesem Zeitpunkt ist das statische Drehmoment (T) so groß wie das dynamische Drehmoment und die Winkelbeschleunigung ist maximal ($\Delta\Theta''_{max}$).

$$I * \Delta\Theta''_{max} = T_{iso} \quad (3.16)$$

$$I = \frac{T_{iso}}{\Delta\Theta''_{max}} \quad (3.17)$$

Aus dem Verhältnis der Änderung der Winkelbeschleunigung $\Delta\Theta''$ und der Änderung der Winkelposition $\Delta\Theta$, multipliziert mit dem entsprechenden Trägheitsmoment I kann nun die musklotendinöse Steifigkeit S der SEC berechnet werden [13].

$$S = \frac{\Delta\Theta''}{\Delta\Theta} * I \quad (3.18)$$

Diese errechneten Steifigkeitswerte werden anschließend mit den isometrischen, vom Probanden erzeugten Drehmomentwerten in Relation gesetzt. So entstehen die QR-typischen Steifigkeits-Drehmoment-Diagramme. Diese Beziehung zwischen Steifigkeit und Drehmoment verläuft in der Regel in einem linearen Verhältnis [13][78].

3.6.2 Python-Automatisierung der Quick Release Auswertung

3.6.2.1 Auswertung per Ableitung des Angular_Position-Datenstreams

Die in Abschnitt 3.6.1 genannten Schritte nach Lambertz et. al. [13] zur Auswertung werden in der Praxis automatisiert mit Hilfe des Python-Skriptes "*mares_QRanalysis4-0.py*" durchgeführt. Der komplette Python-Code zur Auswertung der MARES-Daten befindet sich im Anhang (Anhang A.3) und wurde von Herrn Dr. rer. nat. Uwe Mittag, einem Mitarbeiter der Abteilung für Luft- und Raumfahrtmedizin des DLR verfasst.

Das Python Skript dient sowohl der Auslese der dekodierten Datenstreams, der Erkennung der einzelnen Quick Release-Intervalle, der Analyse der relevanten Messparameter zu diesen Zeitpunkten sowie der Weiterverarbeitung dieser Messparameter und Berechnung der Steifigkeiten aus den einzelnen QR-Intervallen.

Dieses Skript unterscheidet sich zum R-Skript *proc_MARES_mv_v0.2* vor allem durch die automatisierte Erkennung der QR-Intervalle und wurde daher im Verlauf der Arbeit präferiert. Alle ausgelesenen und errechneten Messwerte werden automatisiert in eine CSV-Datei (Tabelle 3.4) übertragen und können anschließend von Hand weiterverarbeitet werden. Weiterhin gibt das Skript zu jedem erkannten QR eine grafische Darstellung der Position des SLA sowie den daraus errechneten Parametern aus (Abbildung 3.8).

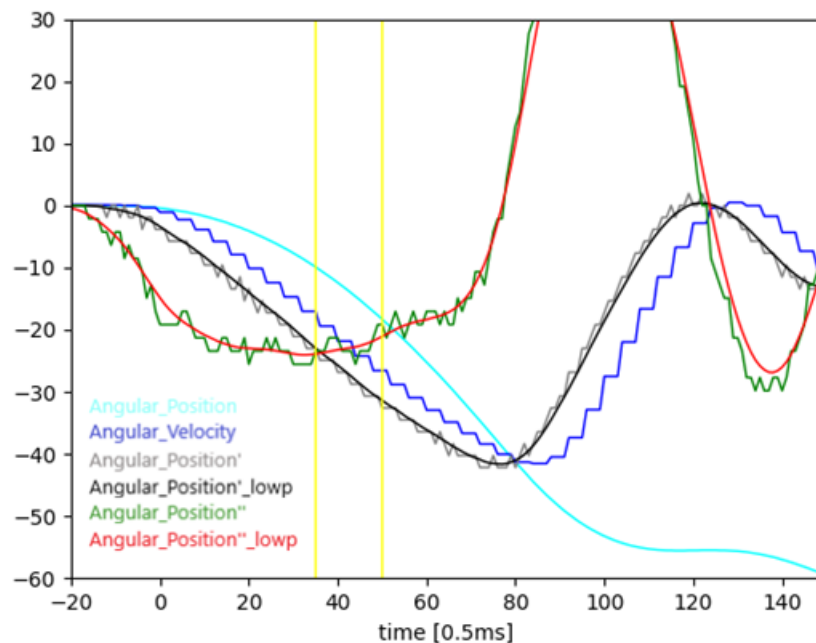


Abbildung 3.8: Darstellung eines vom Python Skript *mares.QRanalysis4 - 0.py* ausgegebenen einzelnen Quick Releases mit den für die Steifigkeitsrechnung relevanten Parametern. Die Skalierung der einzelnen Graphen ist unterschiedlich und wurde zum Zweck einer besseren optischen Vergleichbarkeit angepasst. Quelle: Eigene Abbildung.

Um die in Abschnitt 3.6.1 genannten Parameter zu analysieren, werden die in Abschnitt 3.6 genannten Datenstreams in das Python Skript eingelesen.

Da das MARES ab dem Start des Systems durchgehend Daten aufzeichnet, entsteht ein Datenstream über die Gesamtlaufzeit der Messung, sodass zunächst alle relevanten Zeitsegmente identifiziert werden müssen. Zur automatisierten Identifizierung der QR-Intervalle greift das Python-Skript auf die vom MARES hinterlegten Zeitstempel *Time_Mark_1* und *Time_Mark_2* zu, welche am Start und Ende jedes QR gesetzt werden.

Das Python Skript arbeitet bei der Auswertung grundlegend mit der Winkelposition des short levers *Angular_Position*. Dieser Datenstream wird abgeleitet und anschließend zwei Mal lowpass-gefiltert. So entstehen *Angular_Position'* und *Angular_Position'_lowp*, welche in Abbildung 3.8 in grau bzw. schwarz dargestellt sind.

Dabei entspricht die Ableitung der Winkelposition *Angular_Position'* in grau der vom MARES ausgegebenen blauen Kurve der Winkelgeschwindigkeit *Angular_Velocity* mit einer Latenzzeit von 8 Skalenstrichen (4ms). Die schwarze Kurve für die gefilterte Geschwindigkeit wird ein zweites Mal abgeleitet und man erhält die Winkelbeschleunigung, welche in grün dargestellt ist. Diese wird ebenfalls einmal lowpass-gefiltert und man erhält die in Abbildung 3.8 rot dargestellte Kurve *Angular_Position''_lowp*. Diese Kurve entspricht der Beschleunigung des short levers über den Messzeitraum.

Zur Berechnung der Steifigkeit wird die maximale Beschleunigung nach Auslösen des Quick Release ermittelt. Da sich der SLA beim QR in die definierte Negativrichtung bewegt, wird dazu das Minimum der Beschleunigung im Zeitraum des QRs gesucht. In Abbildung 3.8 ist das Minimum der Beschleunigung durch die erste gelbe Markierung gekennzeichnet.

Zur Berechnung des Trägheitsmomentes I wird das Drehmoment vor Start des QR aus dem Datenstream *Subject_Torque* ausgelesen. Das Trägheitsmoment berechnet sich nach Formel 3.17 aus maximaler Beschleunigung im Messintervall und diesem Drehmoment.

Der Auswertungsbereich beginnt zwei (1 ms) Skalenstriche nach dem Zeitpunkt der maximalen Beschleunigung. Die Änderung der Beschleunigung geteilt durch die Änderung der Winkelposition ($\Delta\theta''/\Delta\theta$) wird über eine Distanz von 15 Skalenstrichen (Ein Skalenstrich entspricht 0,5 ms) ab diesem Zeitpunkt errechnet. Dies wird jeweils um drei Skalenstriche versetzt fünf Mal wiederholt. Dadurch wird ein Bereich von 30 Skalenstrichen (15 ms) abgedeckt. Zusammen mit dem Trägheitsmoment wird die Steifigkeit dieser Werte nach Formel 3.18 berechnet. Aus diesen Steifigkeiten wird ein Mittelwert zusammen mit einer Standardabweichung über diese fünf Werte ausgegeben. Dieser Mittelwert entspricht der Steifigkeit der gemessenen Feder.

Tabelle 3.4: CSV-Ausgabe des Python-Skriptes *mares_QRanalysis4 - 0.py*.

IntervalStart- DateTime(UTC)	IntervalMean- Torque [Nm]	IntervalMaxAng- Acceleration [rad/sec ²]	IntervalMoment- OfInertia [kgm ²]	IntervalDreh- moment [Nm/rad]	IntervalStiff- nessSTD [Nm/rad]
06.10.2020 11:07:45	-67.67	-344.93	0.1962	156.79	21.63
06.10.2020 11:08:33	-62.15	-268.87	0.2311	182.54	14.36
06.10.2020 11:09:20	-60.22	-258.43	0.233	179.27	11.13
06.10.2020 11:10:11	-42.5	-165.54	0.2567	199.6	14.92
06.10.2020 11:10:59	-42.5	-160.22	0.2653	212.89	19.36
06.10.2020 11:11:47	-42.6	-160.64	0.2652	208.53	23.56
06.10.2020 11:12:35	-22.04	-85.65	0.2573	403.39	120.94
06.10.2020 11:13:21	-22.12	-85.22	0.2595	394.26	114.54
06.10.2020 11:14:07	-22.03	-85.22	0.2585	445.9	95.07

In Tabelle 3.4 ist beispielhaft die CSV-Ausgabe des Python Skriptes dargestellt. Alle Parameter, die in der CSV automatisiert errechnet werden, sind im Folgenden erklärt:

- `IntervalStartDateTime(UTC)`: Zeitpunkt zu dem ein einzelner QR durchgeführt wird. Dient der Zuordnung der einzelnen QR Messungen auf den Datenstreams zum geführten Messprotokoll.
- `IntervalMeanTorque [Nm]`: Mittlerer anliegender Drehmoment vor Auslösen des Quick Release
- `IntervalMaxAcceleration [rad/sec2]`: Maximale Winkelbeschleunigung im Messintervall. Sie dient der Berechnung des Trägheitsmoments und damit zur Berechnung der Steifigkeit
- `IntervalMomentOfInertia [kgm2]`: Aus Drehmoment und maximaler Winkelbeschleunigung errechnetes Trägheitsmoment
- `IntervalStiffness [Nm/rad]`: Steifigkeit der Feder im Messintervall die aus Trägheitsmoment, Beschleunigungsdifferenz und Positionsdifferenz im Messintervall bestimmt wird. Es handelt sich um einen Mittelwert der aus fünf verschiedenen Steifigkeitswerten, die in Abständen von drei Skalenstrichen errechnet werden, zusammengesetzt ist.
- `IntervalStiffnessSTD [Nm/rad]`: Standardabweichung der fünf Steifigkeitswerte, welche gemittelt werden, um den ausgegebenen Steifigkeitswert in „IntervalStiffness“ zu erhalten. Messwerte mit hoher Intervall-Standardabweichung sind aus fünf Steifigkeitswerten gemittelt, die größere Schwankungen aufweisen als Steifigkeiten mit einer geringen Intervall-STD. Die Intervall Standardabweichung kann als Maß für Unregelmäßigkeiten der Messung, wie beispielsweise bei Interferenzschwingungen, gesehen werden.

3.6.2.2 Auswertung per Cosinus-Fit

Zusätzlich zur oben genannten Auswertungsmethodik wurde für die Daten ohne Dämpfungseinfluss eine weitere Auswertungsmethode getestet, bei der die Daten mit Hilfe einer Cosinus Funktion gefittet werden. Der Fit erfolgt mittels des Python Skriptes *mares_QRanalysis3-7_delta_55-cosfit.py* (vgl. Anhang A.3). Der Fitbereich beginnt zum gleichen Zeitpunkt wie auch die Auswertungsmethode per Ableitung des Positionstreams. Damit beginnt der Fitbereich ebenfalls zwei Skalenstriche nach der Lage des Maximums der absoluten Beschleunigung im negativen Sinne. Der Fitbereich ist dann 36 Schritte (18 ms) breit. Der Nullpunkt der Fitfunktion ist der Zeitpunkt, zu dem die absolute, gefilterte Geschwindigkeit einen bestimmten Wert überschreitet. Die gelbe Linie in Abbildung 3.9 zeigt die resultierende Cosinus-Funktion, mit Start am gewählten Nullpunkt, wo sich auch das Maximum der Fitfunktion befindet. An dieser Funktion wird anschließend wie in der vorherigen Auswertungsmethode das Trägheitsmoment aus dem Drehmoment und der maximalen Beschleunigung $\Delta\Theta''_{max}$ und damit die Steifigkeit mit Hilfe von $\Delta\Theta''/\Delta\Theta$ errechnet. Die Auswertung erfolgt somit in den Grundzügen gleich wie zuvor, nur

dass nicht der Stream der Winkelposition abgeleitet wird, sondern eine Cosinus-Fit-Funktion die über die Daten gelegt wurde. Die Ergebnisse hierzu finden sich in Abschnitt 4.2.4.

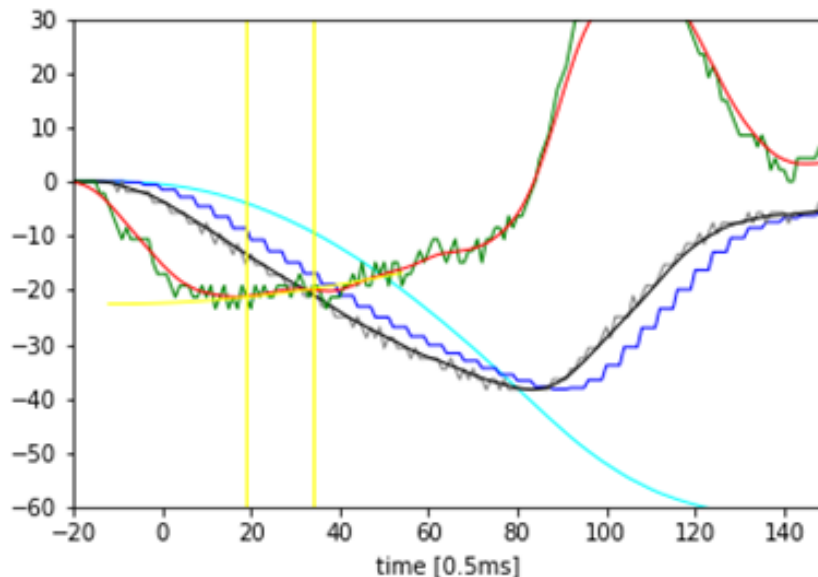


Abbildung 3.9: Darstellung eines vom Python Skript *mares_QRanalysis3 - 7_delta_55 - cosfit.py* ausgegebenen einzelnen Quick Releases mit den für die Steifigkeitsrechnung per Cosinus Fit relevanten Parametern. Die Skalierung der einzelnen Graphen ist unterschiedlich und wurde zum Zweck einer besseren optischen Vergleichbarkeit angepasst. Quelle: Eigene Abbildung.

3.6.3 SP-Python-Automatisierung

Zur Auswertung und Analyse der SP-Messungen wird der Python-Code *bode.py* verwendet. Dieser dient einerseits zur Erzeugung der mathematischen Modellierung nach Kearney et al. gemäß 2.1 und andererseits zur Aufbereitung, Analyse und graphischen Darstellung der experimentell bestimmten Phantom- und AGBRESA-Daten. Zunächst werden hierfür die dekodierten Daten ausgelesen und die darin enthaltenen relevanten Ergebnisse ausgelesen, analysiert und in eine CSV-Datei übertragen. Zur Validierung dieser Rohdaten wurden diese noch händisch überprüft. Dies geschah mittels Prüfung der Übereinstimmung der Datensätze mit den tatsächlich durchgeführten Tests.

3.6.3.1 Verarbeitung der Phantom-Daten

Die zu Anfang des Kapitels angesprochenen, dekodierten Daten werden, zur Vorbereitung für weitere Aufbereitungsschritte, mit einer modifizierten Variante des von Herrn Dr. Uwe Mittag geschriebenen Python-Skriptes *Mares_SP3.py* untersucht. Das Skript liest die MARES-Daten

ein und greift auf die hinterlegten Zeitstempel, sowie Winkelpositions- und Drehmomentsinformationen zu. Sowohl die Winkelpositions- als auch die Drehmomentsinformationen werden mit einem Lowpass-Butterworth-Filter vierter Ordnung gefiltert. Da das MARES ab Start des Systems bis zum Herunterfahren durchgehend Daten aufzeichnet, müssen zunächst alle relevanten Zeitsegmente (also SP-Testdurchführungen) der Gesamtaufzeichnung identifiziert werden. Dies geschieht an Hand der MARES-Zeitstempel, wodurch im Falle der SP-Messmethode der Anfang und das Ende einer Frequenzuntersuchung innerhalb eines SP-Durchlaufs erfasst wird. Mit dieser erfassten Frequenz wird nun weitergearbeitet. Für eine festgelegte Region innerhalb der Frequenz wird jetzt das Drehmoment gemittelt und eine Highpass-Butterworth-Filterung vierter Ordnung für die Drehmoment-Einzeldaten durchgeführt. In einem nächsten Schritt werden die Minima und Maxima der sinusoidalen Winkelpositions- und Drehmomentsverläufe der vorher bestimmten Region erfasst. Zur endgültigen Identifikation der relevanten und validen Daten wird nun einerseits die Frequenz der Sinuskurve der Region bestimmt und überprüft, ob sie in einem geforderten Rahmen liegt (1 bis 20 Hz) und andererseits, ob die Winkelpositions- und Drehmomentsverläufe stimmig zueinander passen, wofür überprüft wird, ob die Differenz zwischen Anzahl der ermittelten Maxima und Minima geringer als sechs ist. Weiterhin werden mit Hilfe der Minima und Maxima der Sinuskurve die einzelnen Winkelpositions- und Drehmomentsamplituden berechnet und anschließend aus den Einzelwerten ein jeweiliger Mittelwert gebildet. Schließlich wird noch die Phasenverschiebung der Region bestimmt und ihr Wert ggf. in einer weiteren Variable normalisiert, um einen Wert der Phasenverschiebung zwischen -360° und 360° zu erhalten. Als Letztes wird durch das Skript eine CSV-Datei mit den Werten für das gemittelte Drehmoment, die gemittelte Drehmomentsamplitude, die Positionsamplitude, die Frequenz, die Phasenverschiebung und die normalisierten Phasenverschiebung erstellt und abgespeichert.

Diese CSV-Datei wird zur anschließenden Verarbeitung mit dem Python-Code ein weiteres mal aufbereitet. Hierfür wird die CSV-Datei händisch um Spalten für den K-, B- sowie MVC-Wert ergänzt. Die eindeutige Zuordnung geschieht an Hand von Testprotokollen sowie an Hand der in den CSV-Dateien enthaltenen Daten.

Die resultierende Datei *MaresSPTable_high.csv* bildet nun die Grundlage für alle weiteren Analyseschritte, welche mit dem Python-Code der *bode.py*-Datei durchgeführt werden. Im ersten Schritt werden mit Hilfe der *readCsv*-Funktion alle Werte aussortiert, die bereits durch *Mares.SP3.py* als Werte außerhalb einer skriptinternen Toleranz erkannt wurden. Auch werden all jene Daten nicht berücksichtigt, die keine Auslenkungsamplitude aufweisen, was sie als fehlerhafte Daten klassifiziert. Anschließend wird, pro Permutation und Frequenz, also reihenweise, die prozentuale Muskelaktivierung aus dem jeweils angegebenen Drehmomentsmittelwert dividiert durch den bekannten MVC errechnet. Hierbei wird eine Toleranzüberprüfung durch die Funktion *roundValue* vorgenommen, um ausschließlich jene %MVC-Werte zu erfassen, die höchstens $\pm 0,05$ von 0; 0,25; 0,5 bzw. 0,75 %MVC abweichen. Ist die Abweichung höher, werden die entsprechenden Daten nicht weiter berücksichtigt. Befindet sich der Wert innerhalb der

Toleranz, wird er anschließend auf 0; 0,25; 0,5 bzw. 0,75 %MVC gerundet.

Da auch die Frequenzwerte Abweichungen aufweisen, werden diese ebenfalls mit der Toleranzüberprüfungsfunktion *roundValue* untersucht. Der Frequenzwert der Messung muss innerhalb einer Toleranz von $\pm 0,3$ Hz des zu erwartenden Frequenzwertes liegen, um weiterhin berücksichtigt zu werden. Ist diese Bedingung erfüllt, wird der Gain als Quotient aus der, in den dekodierten Daten enthaltenen, durchschnittlichen Positionsamplitude in rad/s und der durchschnittlichen Drehmomentsamplitude berechnet und anschließend in dB umgerechnet.

Weiterhin wird die Phase in Grad direkt aus den dekodierten Daten entnommen.

Um diese Werte zuorden- und abrufbar zu machen, werden die Gain- und Phasenwerte mittels eines Dictionarys mit einem eindeutigen Identifikationsschlüssel verknüpft, dies geschieht über die *generateKey*-Funktion.

Im Anschluss werden mit der *buildCsv*- und *writeFreqToCsv*-Funktion die Gain- und Phasenwerte entsprechend ihres Schlüssels in die vorbereitete Output-CSV-Datei eingetragen und die Datei gespeichert.

Im nächsten Schritt werden aus den Gain- und Phasenwerten Bode-Plots erstellt. Diese Plots werden über die *plotting*- bzw. *plotFreqs*-Funktion erzeugt. Diese bilden frequenzabhängig den Gain und die Phase eines Systems ab. Anschließend wird ein Diagramm pro Untersuchungsvariable erstellt und gelabelt, in welchem alle vorhandenen Datensätzen, entsprechend der Ausprägungen der Untersuchungsvariable farblich gruppiert und halbtransparent eingetragen werden. In einem weiteren Schritt wird der Mittelwert der jeweiligen Merkmalsausprägung nicht-transparent und farblich korrelierend in das jeweilige Diagramm eingefügt. Zusätzlich wird ggf. die mathematische Modellierung, welche im folgenden Kapitel thematisiert wird, in die Diagramme eingetragen.

Schließlich werden noch die Standardabweichungen innerhalb der jeweiligen Untersuchungsvariablen berechnet, diese Funktionalität erfüllt die *standardDeviation*-Funktion.

Die vollständigen Codes befinden sich im Anhang.

3.6.3.2 Mathematische Modellierung

Um den Vergleich und somit die Validierung der Testmethoden durchführen zu können, müssen zunächst die rechnerisch zu erwartenden Ergebnisse des Modells erzeugt werden. Der Vergleich der praktischen, realen Messungen mit der Theorie zeigt dann die Validität der zugrundeliegenden Methodik.

Da eine Darstellung des Systemverhaltens über das Frequenzspektrum wünschenswert ist, wird zunächst mit Hilfe einer Laplace-Transformation die Übertragungsfunktion der Differenzialfunktion 2.1 erzeugt, welche anschließend für die Erzeugung der Bode-Plots des mathematisch zu erwartenden Verhaltens genutzt werden. Die Übertragungsfunktion $H(s)$ ist das Verhältnis eines

Ausgangssignals Y zu einem Eingangssignal $T(t)$ als Funktion der komplexen Frequenz s .

$$\begin{aligned}
 T(t) &= I * \frac{d^2\Theta(t)}{dt^2} + B * \frac{d\Theta(t)}{dt} + K * \Theta(t) \\
 \Rightarrow T(t) &= I * s^2 * Y + B * s * Y + K * Y \\
 \Rightarrow T(t) &= Y(I * s^2 + B * s + K) \\
 \Rightarrow H(s) &= \frac{Y}{T(t)} = \frac{1}{I * s^2 + B * s + K}
 \end{aligned} \tag{3.19}$$

Hierbei gilt:

$$\frac{d}{dt} = s, \quad \frac{d^2}{dt^2} = s^2, \quad \theta(t) = Y \tag{3.20}$$

Diese Funktion hat die Form einer Übertragungsfunktion eines PT₂-Gliedes, einem linearen, zeitinvarianten System mit proportionalem Übertragungsverhalten und Verzögerung der zweiten Ordnung. Diese Funktion kann wiederum in eine Normalform gebracht werden. Es ergibt sich:

$$H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{3.21}$$

Aufbauend auf den Prozessen des Unterkapitels 3.6.3.1, werden die zuvor initialisierten Variablen für die Erstellung des rechnerisch zu erwartenden Gesamtsystemverhaltens mit Hilfe der Übertragungsfunktion aus Formel 3.19 als Input verwendet. Dies geschieht im Rahmen der *calcMagPhase*-Funktion über die *signal.lti*- und *signal.bode*-Aufrufe der Python-Bibliothek *scipy*. Weiterhin wird, nach einem Umrechnungsschritt von rad/s in Hz, spezifiziert, dass die Werte der Übertragungsfunktion an den diskreten Frequenzen (4 Hz bis 16 Hz) erfasst werden sollen. Nun werden die, den Frequenzen entsprechenden, Werte für Magnitude und Phase zunächst umgerechnet (Magnitude in Hz und Phase in °) und dann abgespeichert. Die errechneten Werte für Magnitude und Phase werden, wie bei den Phantom-Daten, über die *buildCsv*-Funktion in Abhängigkeit der Frequenzen (4 Hz bis 16 Hz) in die CSV-Datei *bode_plot* geschrieben. Hierbei werden alle Permutationen aus der prozentualen Muskelaktivierung %MVC, der Dämpfung B, der Federkonstanten K und der maximalen Vorspannung MVC berücksichtigt.

Kapitel 4

Ergebnisse

In diesem Kapitel werden die Ergebnisse der im Methodikteil erläuterten Testreihen dargestellt. Zunächst werden hier die Ergebnisse der Ermittlung der Eingangsvariablen dokumentiert. Anschließend werden die Ergebnisse zu den QR- respektive SP-Untersuchungen unter Verwendung des SPQR-Phantoms dargestellt. Schließlich werden noch die AGBRESA-Studiendaten erfasst und Vergleiche der mathematischen Modellierung, der Phantom- und der AGBRESA-Testergebnisse vorgestellt.

4.1 Eingangsparmeter

Zuerst werden nun die Ergebnisse der Charakterisierung der Eingangsparmeter erfasst.

4.1.1 Massenträgheitsmoment

Das Massenträgheitsmoment wurde mittels Drehschwingungsuntersuchung ermittelt. Zunächst wird das theoretische Trägheitsmoment eines Referenzkörpers I_R bestimmt. Der Referenzkörper ist ein kurzer Zylinder aus Stahl mit der Masse $m_R = 5,296 \text{ kg}$ und dem Durchmesser $d_R = 0,1305 \text{ m}$. Nach Formel 3.5 ergibt sich:

$$I_R = \frac{1}{2} * 5,296 \text{ kg} * \left(\frac{0,1305 \text{ m}}{2}\right)^2 = 0,011 \text{ kgm}^2 \quad (4.1)$$

Zur Bestimmung des Eigenträgheitsmomentes der Messvorrichtung I_0 wurde zunächst die Schwingungsdauer eines auf dem Apparatutisch befindlichen Markers ohne Referenzkörper und anschließend mit Referenzkörper gemessen. Nach Einsetzen des Trägheitsmoments des zylindrischen Referenzkörpers $I_R = 0,011 \text{ kgm}^2$, sowie der Schwingungsdauer $T_R = 0,406 \text{ s}$ und $T_0 = 0,290 \text{ s}$ in Formel 3.7 folgt:

$$D_{RM} = \frac{4\pi^2 * 0,011 \text{ kgm}^2}{(0,406 \text{ s})^2 + (0,290 \text{ s})^2} = 5,550 \text{ kgm}^2 \text{ s}^{-2} \quad (4.2)$$

Anschließend lässt sich das Eigenträgheitsmoment I_0 nach Formel 3.4 bestimmen:

$$I_0 = \frac{(0,290\text{ s})^2 * 5,550\text{ kgm}^2\text{s}^{-2}}{4\pi^2} = 0,012\text{ kgm}^2 \quad (4.3)$$

Abschließend kann nun, bei gleichem Vorgehen mit dem eigentlich zu untersuchenden Körper statt des Referenzkörpers, das Massenträgheitsmoment des zu untersuchenden Gesamtsystems gemäß Formel 3.9 bestimmt werden:

$$I = \frac{(0,787\text{ s})^2 * 5,550\text{ kgm}^2\text{s}^{-2}}{4\pi^2} - 0,012\text{ kgm}^2 = 0,075\text{ kgm}^2 \quad (4.4)$$

Im letzten Schritt muss nun die Verschiebung des Körpers mit Hilfe des Satzes von Steiner berücksichtigt werden, da die Drehachse unseres Systems nicht im Schwerpunkt dessen liegt. Die Verschiebung bzw. Distanz zwischen Drehachse und Schwerpunkt wurde gemessen und beträgt $d = 0,148\text{ m}$. Weiterhin fließt das Gewicht des Gesamtsystems ein, welches einen Wert von $m = 6,745\text{ kg}$ hat. Das Massenträgheitsmoment des Gesamtsystems unter Berücksichtigung der Verschiebung lässt sich dann mit Hilfe der Formel 3.10 wie folgt ermitteln:

$$I_v = 0,075\text{ kgm}^2 + 6,745\text{ kg} * (0,148\text{ m})^2 = 0,223\text{ kgm}^2 \quad (4.5)$$

4.1.2 Dämpfungskonstante

Ein weiterer, zentraler Bestandteil der Berechnung des zu erwartenden Phantomverhaltens sind die Dämpfungseigenschaften, bzw. genauer, die rotatorische Dämpfungskonstante. Die rotatorische Dämpfungskonstante lässt sich anhand Formel 3.13 berechnen. Hierfür wurden zunächst nach den Formeln aus Gleichung 3.11 F und v berechnet:

$$F = 1,8\text{ kg} * 9,81\frac{\text{m}}{\text{s}^2} = 17,7\text{ N} \quad (4.6)$$

$$v = \frac{0,05\text{ m}}{5\text{ s}} = 0,01\frac{\text{m}}{\text{s}} \quad (4.7)$$

Anschließend konnten gemäß Gleichung 3.12 M sowie ω , unter Berücksichtigung des gemessenen Hebelarms $r = 0,238\text{ m}$, errechnet werden:

$$M = 17,7\text{ N} * 0,238\text{ m} = 4,2\text{ Nm} \quad (4.8)$$

$$\omega = \frac{v}{r} = \frac{0,01\frac{\text{m}}{\text{s}}}{0,238\text{ m}} = 0,04\frac{\text{rad}}{\text{s}} \quad (4.9)$$

Schließlich ergibt sich aus 3.13 für die rotatorische Dämpfungskonstante B :

$$B = \frac{4,2\text{ Nm}}{0,04\frac{\text{rad}}{\text{s}}} = 105\text{ Nmsrad}^{-1} \quad (4.10)$$

4.1.3 Drehfederkonstanten

Abschließend wurden noch die Herstellerangaben bezüglich der Federkonstanten überprüft. Hierbei wurde gemäß Formel 3.14 die Steigung der Geradengleichung der Auslenkung in Abhängigkeit der resultierenden Kraft errechnet. Nachstehend, in Tabelle 4.1, sind die Daten der Validierung zu sehen.

Tabelle 4.1: Ergebnisse der Federkonstantenvalidierung. Die vier Federn wurden ausgelenkt. Diese Auslenkung und die daraus resultierende Kraft wurden über eine Kofferwaage ermittelt. Die Tabelle stellt die Daten dieser Messdatenerfassung dar.

Federkonstante laut Hersteller [N/mm]	Auslenkung der Feder [mm]	Resultierende Kraft [N]	Ermittelte Federkonstante [N/mm]	Bestimmtheitsmaß	Abweichung der Federkonstanten [%]
1,48	156	280	1,37	1,0000	7
	199	339			
	222	370			
2,89	152	498	2,14	0,8902	26
	177	552			
	165	510			
4,55	81	464	4,57	0,9999	0
	82	470			
	106	579			
8,76	36	464	8,52	0,9999	3
	47	559			
	54	617			

In der folgenden Abbildung sind die Federn entsprechend ihrer Federkonstanten laut Hersteller bezeichnet. Aufgetragen ist die resultierende Kraft in Abhängigkeit der Auslenkung der jeweiligen Federn.

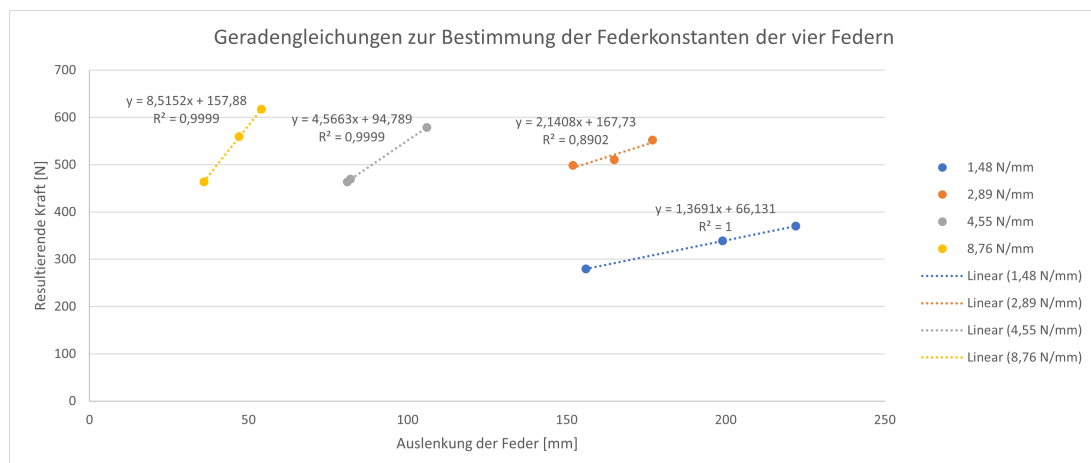


Abbildung 4.1: Darstellung der Messung von resultierender Kraft in Abhängigkeit der Federauslenkung zur Validierung der Federkonstanten laut Hersteller. In der Abbildung sind die Geradengleichungen der Federn inklusive ihres Bestimmtheitsmaßes angegeben. Die Federn sind anhand ihrer Federkonstanten laut Hersteller bezeichnet.

Die Umrechnung der translatorischen Federkonstanten in Drehfederkonstanten erfolgte nach Formel 3.15: $K_1 = 80 \text{ Nm/rad}$, $K_2 = 158 \text{ Nm/rad}$, $K_3 = 248 \text{ Nm/rad}$ und $K_4 = 478 \text{ Nm/rad}$.

4.2 Quick Release

Dieses Kapitel behandelt die Ergebnisse der QR-Testmethoden für das SPQR-Phantom und die AGBRESA-Daten.

Steifigkeiten mit einer Intervall-Standardabweichung von >500 (vgl. Abschnitt 3.6.2) wurden in folgenden Darstellungen nicht berücksichtigt. Die entsprechenden Messungen zeigten starke Interferenzvibrationen oder andere Fehler, wie etwa das sich Lösen von Schrauben durch starke Schwingungen. Dies wurde im Testprotokoll während der Messung dokumentiert und entsprechend markierte Messungen weisen oft die zu hohen Intervall-Standardabweichungen auf.

Die Auswertung der Ergebnisse für die QR-Messungen fokussiert sich primär auf die Messungen ohne parallel geschalteten Dämpfer, da während der Versuche festgestellt wurde, dass der Dämpfer durch eine zu hoch gewählte Dämpfung einen zu starken Einfluss auf die Messwerte der Steifigkeit ausübt und der Einfluss der Federkonstante auf die Ergebnisse nicht mehr ersichtlich ist. Die Messungen mit der vorherigen Position der Endstops wurde komplett verworfen.

Insgesamt wurden 144 einzelne Quick Release Tests durchgeführt. Von diesen 144 einzelnen QRs wurden 137 durch das Python Skript erkannt und weiterverarbeitet, wovon wiederum 121 eine Intervall-Standardabweichung von <500 aufweisen und damit für die Auswertung berücksichtigt werden.

Diese 121 QRs lassen sich in 64 QR Messungen ohne einen Dämpfer und 57 QR Messungen mit Einfluss eines Dämpfers trennen.

Tabelle 4.2: Anzahl der gemessenen Quick Release Messungen mit der Anzahl der durch das Python Skript detektierten QRs sowie Anzahl der Messungen mit annehmbaren Intervall-STDs.

Konfiguration	keine Dämpfung	Dämpfung	Summe
Anzahl gemessener QRs	72	72	144
Anzahl detektierter QRs	71	66	137
Anzahl verwerteter QRs	64	57	121

4.2.1 Bestimmung der Federsteifigkeit K

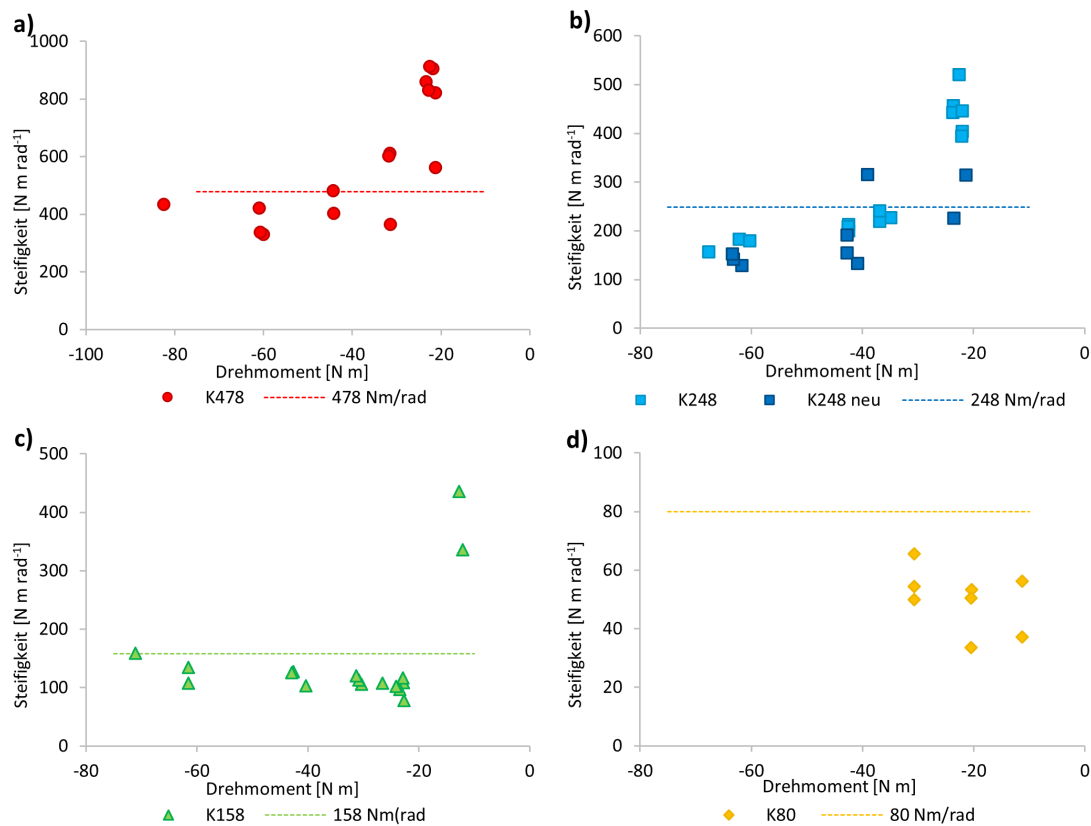


Abbildung 4.2: Darstellung der Relation von ermittelter Steifigkeit zu angelegtem Drehmoment der Federn a) K_{478} , b) K_{248} und $K_{248\text{neu}}$, c) K_{158} und (d) K_{80} . In vorliegender Darstellung wurden die QRs ohne Einfluss eines Dämpfers aufgenommen. Die Drehfederkonstante der jeweiligen Feder ist als Gerade eingezeichnet.

Abbildung 4.2 zeigt die Relation von ermittelter Steifigkeit zu angelegtem Drehmoment für die vier vermessenen Federn a) K_{478} , b) K_{248} , c) K_{158} und d) K_{80} ohne den Einfluss eines parallel geschalteten Dämpfers in QR-typischen Steifigkeit-Drehmoment Diagrammen. Abbildung 4.2b zeigt zudem die ermittelten Steifigkeiten für K_{248} nachdem der voreingestellte Trägheitsmoment von $0,35 \text{ kgm}^2$ zu $0,097 \text{ kgm}^2$ geändert wurde ($K_{248\text{neu}}$).

Die Drehfederkonstante der einzelnen Federn ist mittels einer Geraden eingezeichnet und entspricht der errechneten wahren Steifigkeit der entsprechenden Feder.

Abbildung 4.2a zeigt für die Feder K_{478} bei angelegten Drehmomenten von über 45 Nm Steifigkeitswerte unter den zu erwartenden 478 Nm/rad^{-1} . Bei geringeren Drehmomenten als 45 Nm liegen die Werte primär über der zu erwartenden Steifigkeit. Bei einem Drehmoment von 20 Nm wurde die höchste Steifigkeit mit 905 Nm/rad^{-1} gemessen. Die Höhe des angelegten Drehmoments korreliert stark mit der Höhe der ermittelten Steifigkeit ($r = 0.74$, $p < 0.01$).

Die Steifigkeiten für K_{248} verlaufen bei höheren angelegten Drehmomenten als 40 Nm ebenfalls unter dem Erwartungswert für die Federsteifigkeit. Bei Drehmomenten von mehr als 40 Nm lie-

gen die Werte primär über dem Erwartungswert für beide voreingestellten Trägheitsmomente. Sowohl für K_{248} ($r = 0.86$, $p < 0.01$) als auch für K_{248} neu ($r = 0.71$, $p < 0,05$) korrelieren Drehmoment und Steifigkeit stark, wenn auch mit unterschiedlichen Signifikanzniveaus.

Die beiden Federn K_{478} und K_{248} mit den jeweils größten Drehfederkonstanten zeigen beide einen Aufwärtstrend der ermittelten Steifigkeit mit sinkendem angelegtem Drehmoment in einem signifikanten Ausmaß.

Abbildung 4.2c zeigt die Steifigkeiten der Feder K_{158} . Bis auf zwei Werte variieren die Steifigkeiten zwischen 78 und 158 Nmrad^{-1} . Im Bereich des kleinsten angelegten Drehmoments ($25\% \text{ MVC} \approx 10 \text{ Nm}$) wurden zwei Steifigkeiten gemessen, die über dem erwarteten Wert von 158 Nmrad^{-1} liegen. Diese weisen damit die größte Abweichung von der Drehfederkonstante der Feder auf. Auch hier tritt wie bei 4.2a) und b) die höchste ermittelte Steifigkeit bei Drehmomenten im kleinsten gemessenen Bereich auf. Die Steifigkeiten korrelieren moderat mit dem angelegten Drehmoment ($r = 0.34$, $p > 0.05$), sind jedoch nicht signifikant. Die ermittelten Steifigkeiten ohne die zwei höchsten Messwerte korrelieren stark positiv ($r = 0.72$, $p < 0,01$).

Abbildung 4.2d zeigt die Feder K_{80} mit der geringsten Drehfederkonstante von 80 Nmrad^{-1} . Alle ermittelten Steifigkeiten sind kleiner als die Drehfederkonstante der Feder. Das Maximum der Steifigkeitswerte liegt hier mit 65 Nmrad^{-1} bei einem angelegten Drehmoment von 35 Nm . Der kleinste Wert tritt mit 34 Nmrad^{-1} bei 21 Nm auf. Die Steifigkeiten korrelieren moderat negativ mit dem angelegten Drehmoment, sind jedoch nicht signifikant ($r = -0.45$, $p > 0.05$).

Tabelle 4.3: Arithmetisches Mittel der Steifigkeiten aller Federn sortiert nach Drehmomenten die als Vorspannung angelegt wurden.

Moment [Nm]	S K_{80} [Nmrad ⁻¹]	S K_{158} [N m rad ⁻¹]	S K_{248} (0,35 kgm ²) [N m rad ⁻¹]	S K_{248} (0,097 kgm ²) [N m rad ⁻¹]	S K_{478} [N m rad ⁻¹]
10	47	386	/	/	/
20	46	102	444	285	815
30	57	113	229	/	527
40	/	118	207	160	442
60	/	134	173	141	363
Durchschnitt	50	146	299	195	592

Tabelle 4.3 zeigt eine Zusammenfassung der Steifigkeiten aus Abbildung 4.2 gemittelt für die verschiedenen Drehmomentbereiche die angelegt wurden. Feder K_{158} zeigt die höchste Steifigkeit bei 30 Nm , alle anderen Federn weisen die höchste Steifigkeit bei dem niedrigsten vermessenen Drehmoment auf. Es ist zu beachten, dass unterschiedliche Bereiche aus unterschiedlich vielen Messwerten bestehen. Der 20 Nm Bereich wird beispielsweise sowohl durch $50\% \text{ MVC}$ einer 40 Nm Maximalkraft als auch durch $25\% \text{ MVC}$ einer 80 Nm Maximalkraft erreicht, der 10 Nm Bereich lediglich durch $25\% \text{ MVC}$ des 40 Nm MVC .

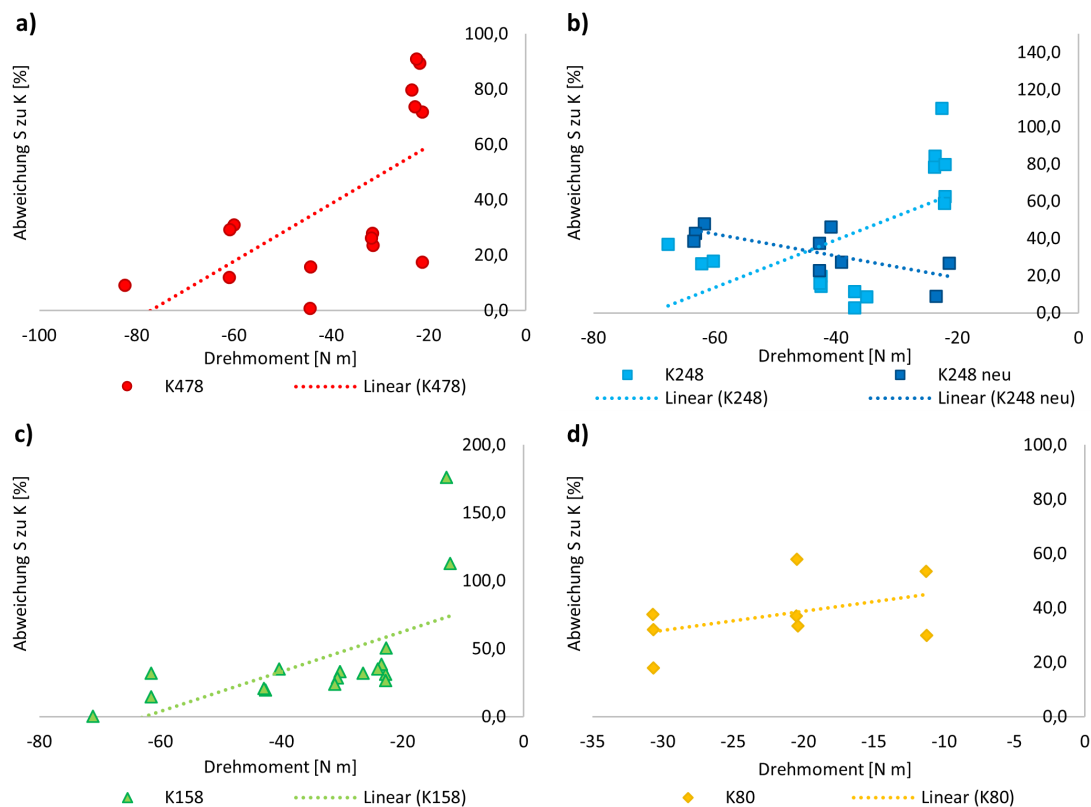


Abbildung 4.3: Prozentuale Abweichung der Federsteifigkeiten von den ermittelten Drehfederkonstanten von a) K_{478} , b) K_{248} und $K_{248\text{neu}}$, c) K_{158} und d) K_{80} in Relation zum angelegten Drehmoment.

In Abbildung 4.3 ist die prozentuale Abweichung der in Abbildung 4.2 dargestellten ermittelten Steifigkeiten von der Drehfederkonstante der Federn a) K_{478} , b) K_{248} , c) K_{158} und d) K_{80} zu sehen. Die Abweichungen sind dabei als Betrag dargestellt und mit Trendlinien versehen. Alle Federn mit dem ursprünglich für QR voreingestellten Trägheitsmoment weisen eine Entwicklung zu hohen Abweichungen bei kleinen am MARES anliegenden Drehmomenten auf. Feder K_{248} mit dem verringertem Trägheitsmoment von $0,097 \text{ kgm}^2$ weist einen umgekehrten Trend auf und es ergeben sich bei hohen Drehmomenten hohe Abweichungen, die mit kleiner werdendem Moment sinken.

Die Steifigkeitswerte für das kleiner voreingestellte Trägheitsmoment von $0,097 \text{ kgm}^2$ weichen dadurch bei kleinen Drehmomenten weniger vom Erwartungswert ab als die Werte für das ursprüngliche Trägheitsmoment von $0,35 \text{ kgm}^2$.

Tabelle 4.4: Arithmetisches Mittel der Abweichungen der Steifigkeiten von der Drehfederkonstante aller Federn sortiert nach Drehmomenten die als Vorspannung angelegt wurden.

Moment [Nm]	$\Delta K_{80}[\%]$	$\Delta K_{158}[\%]$	$\Delta K_{248}(0,35\text{ kgm}^2)[\%]$	$\Delta K_{248}(0,097\text{ kgm}^2)[\%]$	$\Delta K_{478}[\%]$	$\Delta K_{\text{gesamt}}[\%]$
10	42	144	/	/	/	93
20	43	36	79	21	70	50
30	29	28	8	/	26	23
40	/	25	17	36	8	21
60	/	16	30	43	24	28
Durchschnitt	37	42	42	33	40	39

Tabelle 4.4 zeigt die durchschnittlichen prozentualen Abweichungen der ermittelten Steifigkeit der Federn von ihrer entsprechenden Drehfederkonstante aus Abbildung 4.3. Die durchschnittliche Abweichung der Federn liegt bei allen Federn zwischen 30 und 42 %. Der Durchschnitt über alle Federn bei allen Drehmomenten liegt bei 39 %. Feder K_{80} weist die kleinste Abweichung bei einem angelegten Moment von 30 Nm auf. Alle drei anderen Federn weisen die höchste Abweichung beim niedrigsten angelegten Moment. Feder K_{248n} zeigt eine umgekehrte Tendenz mit der höchsten Abweichung vom Erwartungswert beim höchsten gemessenen Moment. Insgesamt ist K_{248n} mit verändertem Trägheitsmoment auch die Feder mit der geringsten durchschnittlichen Abweichung vom Wert der Drehfederkonstante. Betrachtet man die durchschnittlichen Abweichungen aller Federn getrennt nach Drehmomentsbereichen zeigt sich, dass die geringsten Abweichungen bei Drehmomenten zwischen 30 und 40 Nm mit jeweils 23 % und 21 % auftreten. Die höchste Abweichung tritt durchschnittlich bei einem Drehmoment von 10 Nm mit 93 % auf.

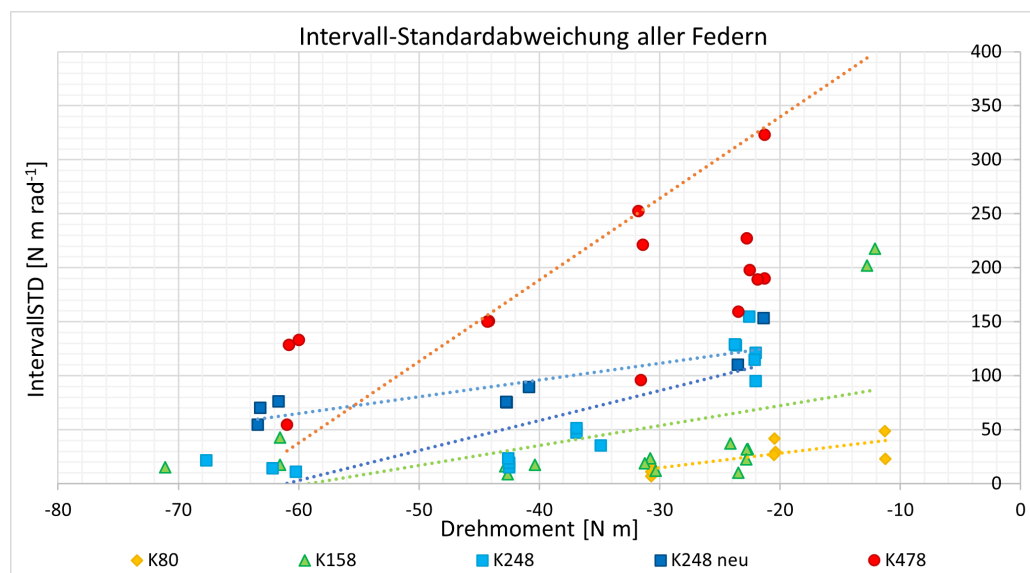


Abbildung 4.4: Intervall-Standardabweichung der vier vermessenen Federn im Vergleich zu dem angelegten Drehmoment ohne Einfluss eines Dämpfers. Es sind Trendlinien der jeweiligen Datensätze eingezeichnet.

Abbildung 4.4 zeigt die Intervall-Standardabweichungen (vgl. Abschnitt 3.6.2) zu allen Steifigkeitswerten der vier vermessenen Federn mit eingezeichneten Trendlinien. Alle Trendlinien weisen eine Tendenz zu hohen Intervall-Standardabweichungen bei niedrigen Drehmomenten auf. Es wurde die Korrelation von Drehmoment und Intervall-Standardabweichungen untersucht. Die Intervall-STD für K_{478} korreliert stark mit dem angelegten Drehmoment ($r = 0.704$, $p < 0.01$). Für K_{248} zeigt sich ebenfalls eine starke Korrelation ($r = 0.84$, $p < 0.01$) beider Faktoren. Für K_{248} neu ist die starke Korrelation nicht signifikant ($r = 0.607$, $p > 0.05$). Die Intervall-STD für K_{158} korreliert moderat mit dem angelegten Drehmoment ($r = 0.49$, $p < 0.05$) und für K_{80} zeigt sich wieder eine starke Korrelation zwischen Intervall-STD und Drehmoment ($r = 0.755$, $p < 0.05$).

Um die Streuung der Messwerte bei mehrfacher Messung mit gleichen Messparametern (%MVC, Drehfederkonstante) zu untersuchen, wurde der Variationskoeffizient der Steifigkeit von den jeweils drei aufeinanderfolgenden QR Messungen mit den gleichen Messparametern ermittelt und über dem angelegtem Drehmoment aufgetragen (Abbildung 4.5).

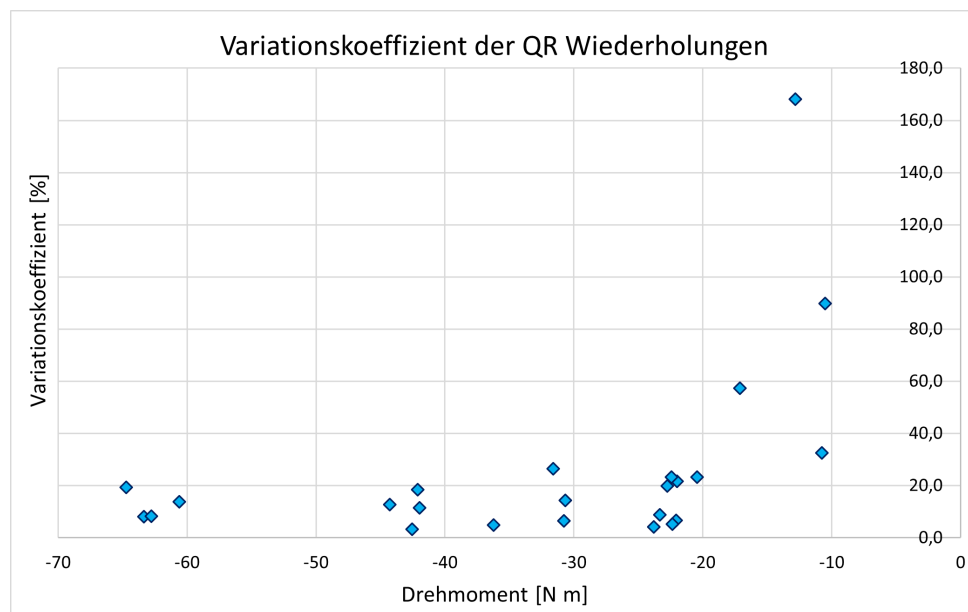


Abbildung 4.5: Variationskoeffizient aller Federn in Relation zum angelegten Drehmoment ohne Einfluss eines Dämpfers.

Bei den Messungen der geringsten Drehmomente zwischen 10 und 20 Nm sind mit Werten zwischen 30 % und 170 % die höchsten Variationskoeffizienten und damit die höchsten Schwankungen bei wiederholten Messungen zu sehen. Bei Drehmomenten von über 20 Nm schwanken alle Variationskoeffizienten zwischen 3 % und 27 %. Es gibt eine moderate Korrelation zwischen angelegtem Drehmoment und dem Variationskoeffizienten der Steifigkeitswerte der QR Messungen ($r = 0.442$, $p < 0.05$). Mit geringerem Drehmoment gibt es also größere Unterschiede der ermittelten Steifigkeiten bei wiederholten Messungen.

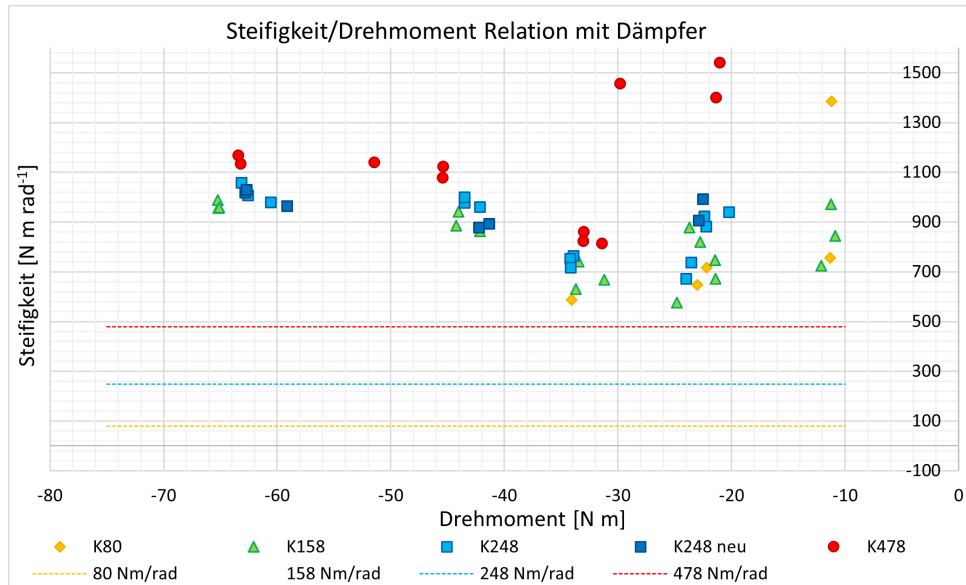


Abbildung 4.6: Darstellung der Relation von ermittelter Steifigkeit und angelegtem Drehmoment der Federn K_{478} (rot), K_{158} (grün), K_{80} (gelb) und K_{248} sowie K_{248} mit zwei unterschiedlich eingestellten Trägheitsmomenten (hell- und dunkelblau).

Die Steifigkeiten der Federn K_{478} , K_{248} , K_{158} und K_{80} für QR-Messungen mit parallel geschaltetem Dämpfer sind in Abbildung 4.6 dargestellt. Die Steifigkeiten sind für alle Federn höher als für die QR-Messungen ohne Dämpfer. Die geringste gemessene Steifigkeit liegt bei 575 Nmrad^{-1} , die höchste bei 1540 Nmrad^{-1} . Damit bewegen sich alle Messwerte über dem Erwartungswert für die Feder mit der höchsten Drehfederkonstante.

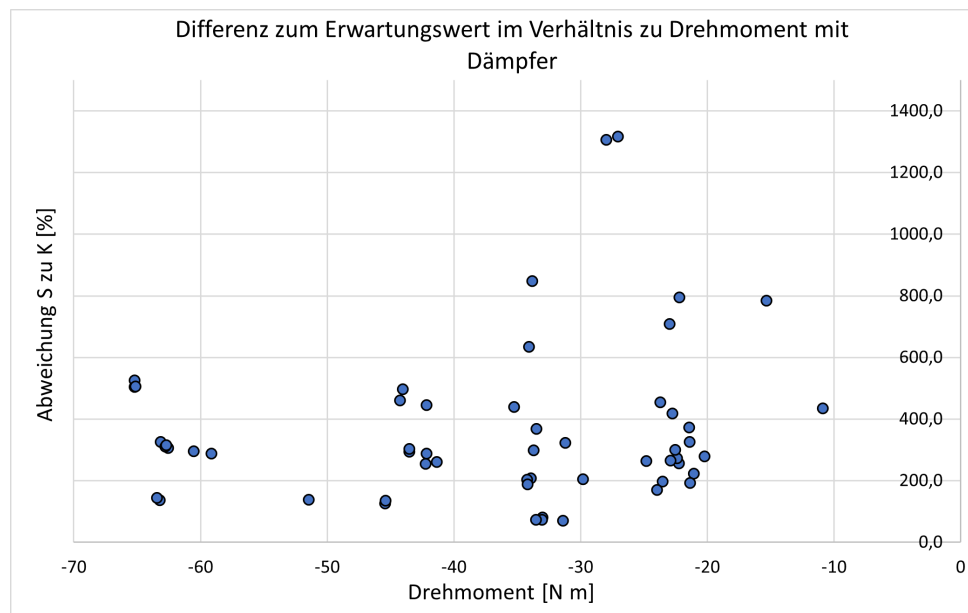


Abbildung 4.7: Prozentuale Abweichung der Federsteifigkeiten von den ermittelten Drehfederkonstanten der einzelnen Federn in Relation zum angelegten Drehmoment mit Einfluss eines Dämpfers.

In Abbildung 4.7 sind die zu Abbildung 4.6 gehörenden prozentualen Abweichungen aller Federn vom Erwartungswert dargestellt. Alle Steifigkeiten weichen positiv ab mit einer minimalen Abweichung von 70 %. Generell liegen alle Messwerte zwischen 70 % und 1300 % Abweichung, was deutlich höhere Abweichungen als bei den Messungen ohne Dämpfer darstellt.

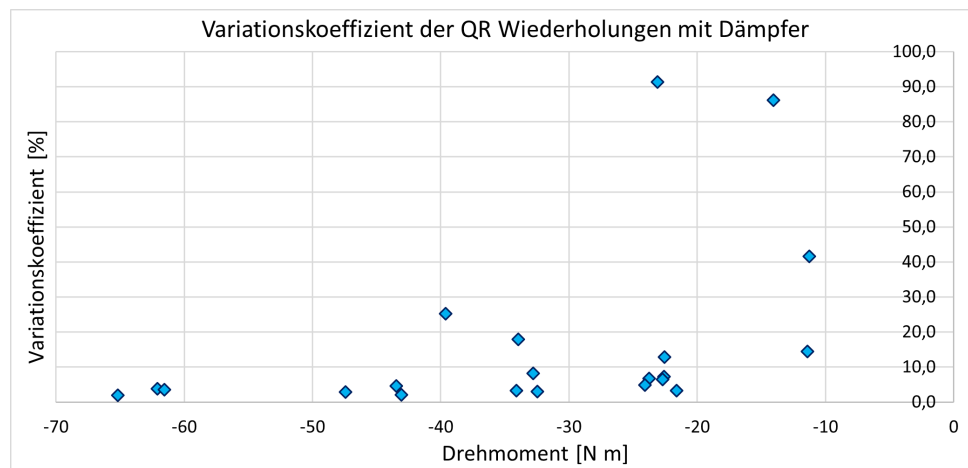


Abbildung 4.8: Variationskoeffizient aller Federn in Relation zum angelegten Drehmoment mit Einfluss eines Dämpfers.

Abbildung 4.8 zeigt die Variationskoeffizienten aller Steifigkeiten aller Federn mit Dämpfung. Wie auch bei ungedämpften QR-Messungen besteht eine moderate Korrelation zwischen angelegtem Drehmoment und dem Variationskoeffizienten der Messwerte ($r = 0.445$, $p < 0.05$).

4.2.2 Einfluss des voreingestellten Trägheitsmoment im MARES

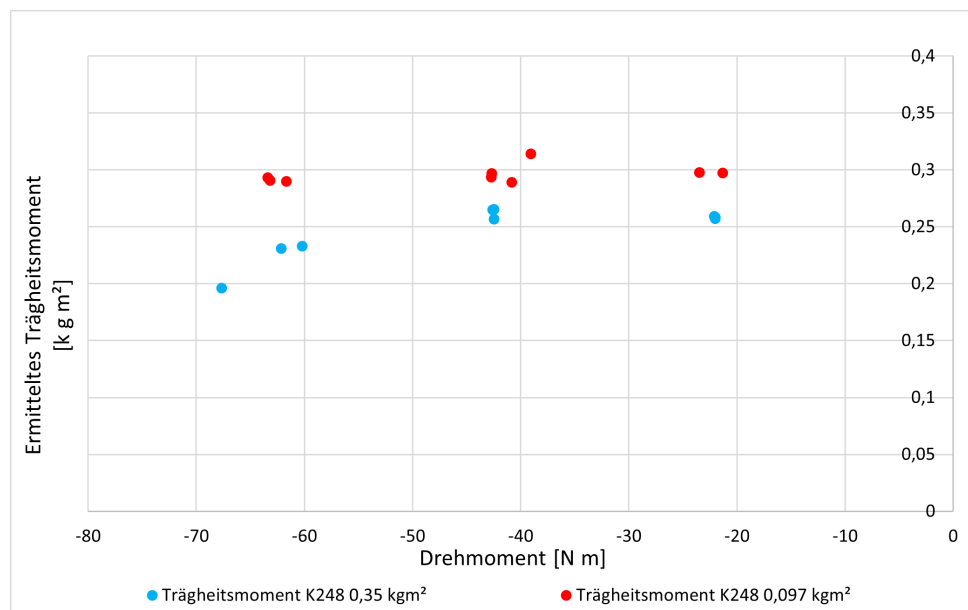


Abbildung 4.9: Darstellung der ermittelten Trägheitsmomente für das alte voreingestellte Trägheitsmoment von $0,35 \text{ kgm}^2$ (blau) und das neue voreingestellte Trägheitsmoment von $0,097 \text{ kgm}^2$ (rot).

Die QR-Messreihe der Feder K_{248} wurde mit einem geänderten voreingestellten Trägheitsmoment des SLA und anhängendem Phantom wiederholt. In Abbildung 4.9 sind die vom Python Skript anhand von Winkelbeschleunigung und Drehmoment errechneten Trägheitsmomente dieser Messreihe dargestellt (vgl. Formel 3.17). Die Messreihe des für AGBRESA genutzten Moments von $0,35 \text{ kgm}^2$ ist in blau dargestellt, die Messreihe mit einem geringeren Trägheitsmoment von $0,097 \text{ kgm}^2$ in rot. Alle anderen Messparameter der beiden Messreihen sind gleich.

Der Mittelwert der errechneten Trägheitsmomente der Messung mit einem voreingestellten Moment von $0,35 \text{ kgm}^2$ beträgt $(0,247 \pm 0,023) \text{ kgm}^2$. Der Mittelwert der Messung mit einem voreingestellten Trägheitsmoment von $0,097 \text{ kgm}^2$ beträgt $(0,296 \pm 0,008) \text{ kgm}^2$.

4.2.3 Muskelsteifigkeiten der AGBRESA-Studie

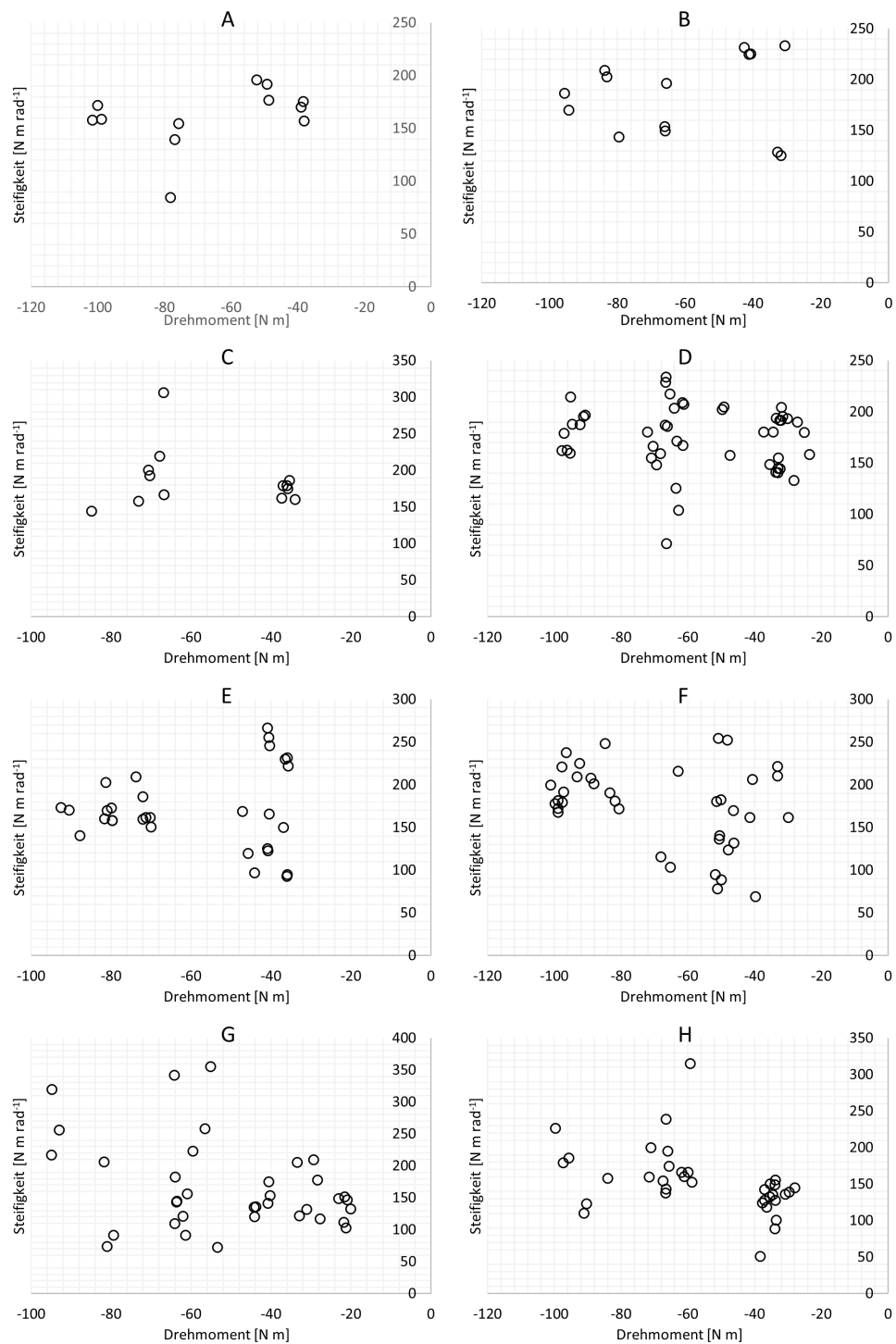


Abbildung 4.10: Darstellung der Steifigkeit-Drehmoment-Relation für die ausgewählten AGBRESA Probanden A-H.

Abbildung 4.10 zeigt die Ergebnisse von acht AGBRESA Probanden (A-H), welche im Laufe der AGBRESA Studie Quick Release Tests zur muskulotendinösen Steifigkeitsbestimmung durchgeführt haben. Die Steifigkeit wurde dabei mit der in Abschnitt 3.6.2 beschriebenen Methodik untersucht.

Für alle Probanden wurde die Korrelation von angelegtem Drehmoment zu ermittelter Steifigkeit untersucht. Die Ergebnisse zur Korrelation sowie die Mittelwerte für die Steifigkeiten der einzelnen Probanden ist in Tabelle 4.5 dargestellt.

Tabelle 4.5: Mittelwert mit Standardabweichungen der AGBRESA Probanden A-H mit jeweiliger Korrelation der Steifigkeiten zum angelegten Drehmoment.

Proband	Steifigkeit [N m rad ⁻¹]	Korrelationskoeffizient Pearson
A	202 ± 68	r = -0.07, p >0.05
B	197 ± 49	r = -0.06, p >0.05
C	213 ± 74	r = -0.12, p >0.05
D	202 ± 49	r = 0.23, p >0.05
E	173 ± 48	r = 0.02, p >0.05
F	183 ± 57	r = -0.29, p >0.05
G	186 ± 83	r = -0.46, p <0.01
H	188 ± 76	r = -0.59, p <0.01

Die Probanden A-F weisen entweder keine oder eine schwache Korrelation von Drehmoment zu Steifigkeit auf, dabei ist keines der Ergebnisse signifikant. Die Probanden G und H weisen eine moderate (G) und eine starke (H) negative Korrelation von Drehmoment zu Steifigkeit auf. Beide Ergebnisse sind auf einem Niveau von 0.01 signifikant.

Aufgrund der in den meisten Fällen fehlenden Korrelation von Drehmoment zu Steifigkeit wurden die Steifigkeitswerte der Daten nicht wie in Tabelle 4.3 nach Drehmomentsbereichen unterteilt. Die Mittelwerte der Steifigkeit aller Probanden liegen zwischen 173 und 202 Nmrad⁻¹.

4.2.4 Federsteifigkeiten mit Auswertung durch Cosinus-Fit-Annäherung

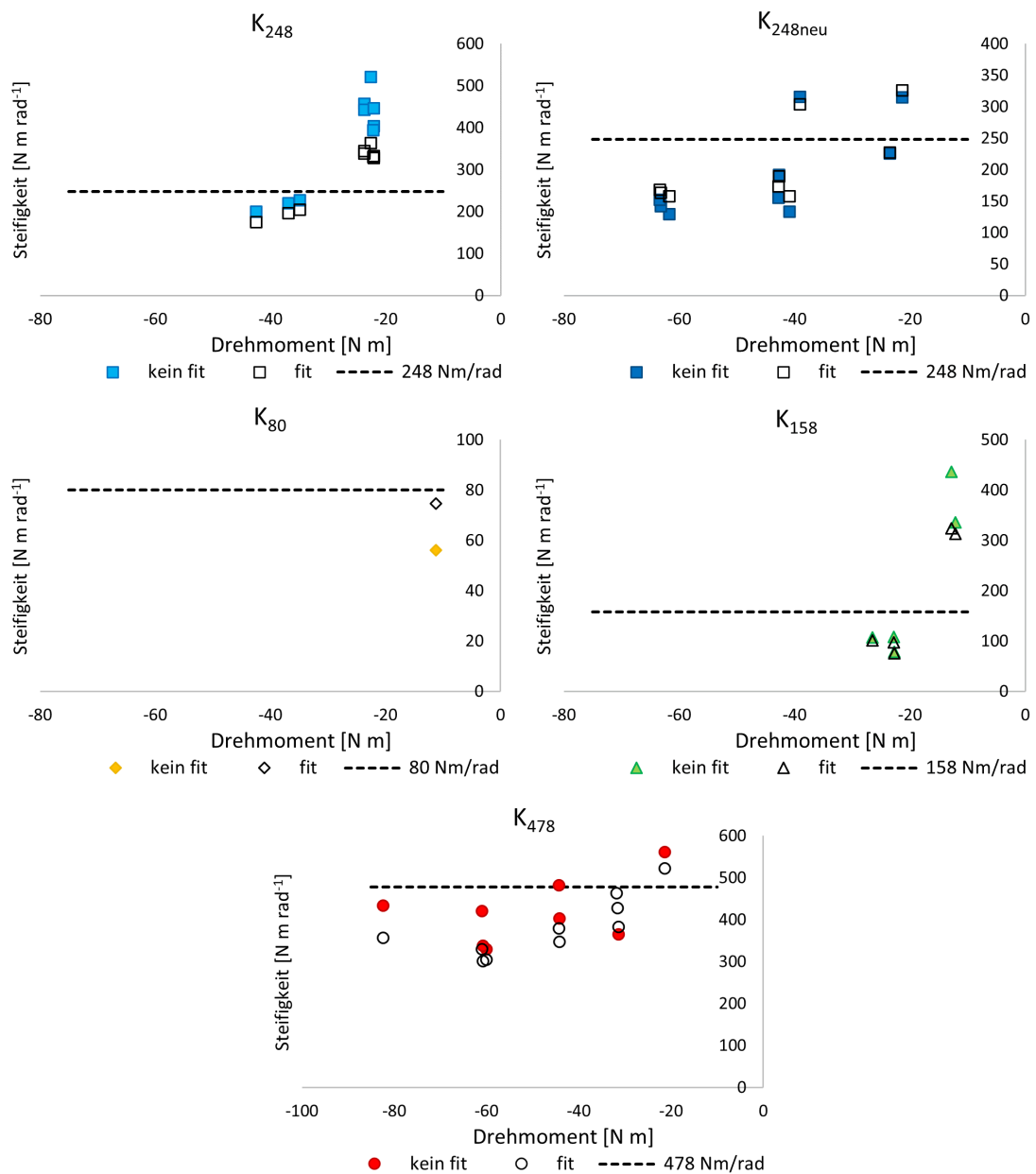


Abbildung 4.11: Vergleich der ermittelten Steifigkeiten der vier vermessenen Federn K_{478} , K_{248} , K_{158} und K_{80} nach Auswertung mit herkömmlicher Auswertungsmethodik (farbig) und nach neuer Fit-Methode (weiß).

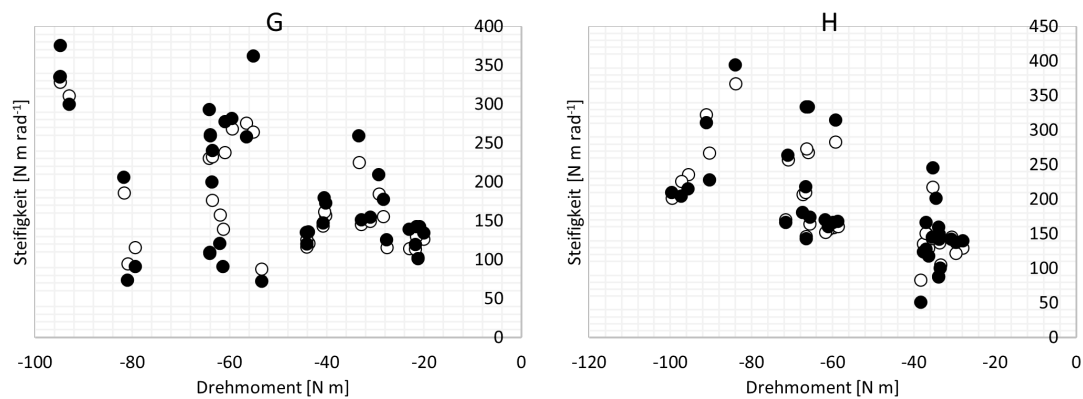


Abbildung 4.12: Vergleich der ermittelten Steifigkeiten der AGBRESA-Probanden G und H nach Auswertung mit herkömmlicher Auswertungsmethodik (schwarz) und nach neuer Fit-Methode (weiß).

Die Auswertung per Cosinus-Fit lieferte nicht bei allen Messwerten einen Erfolg und der Fit konnte sich bei einem Teil der Daten nicht an den Graph zur Beschleunigung anlegen, sodass weniger Messwerte zur Verfügung stehen als bei den normal ausgewerteten Messungen.

Die Ergebnisse der Auswertung der Daten nach der Auswertung mit einer Cosinus-Fit-Funktion sind in Abbildung 4.11 dargestellt. Die Steifigkeiten der Fit-Methodik mit Ableitung der Winkelposition sind, wie in vorherigen Ergebnissen, farbig kodiert. Die Ergebnisse der Cosinus-Fit-Methode sind in weiß dargestellt. Zusätzlich wurden die Probanden G und H aus der AGBRESA Studie nach der Cosinus-Fit-Methode ausgewertet. Diese Daten wurden in Abbildung 4.12 dargestellt. Die Daten der herkömmlichen Auswertung per Ableitung des Positionstream sind hier schwarz kodiert, die Daten der Ableitung per Ableitung des Cosinus-Fits sind weiß dargestellt. Die Mittelwerte beider Auswertungsmethoden für die vier Federn sowie die AGBRESA Probanden finden sich in Tabelle 4.6.

Tabelle 4.6: Mittlere Steifigkeiten der vermessenen Federn K_{478} , K_{248} , K_{158} und K_{80} und der AGBRESA Probanden G und H nach Auswertung mit herkömmlicher Auswertungsmethodik (Ableitung AngPos) und nach neuer Fit-Methode (Ableitung Cos-Fit).

	Ableitung AngPos	Ableitung Cos-Fit
S K_{80} [N m rad ⁻¹]	56	75
S K_{158} [N m rad ⁻¹]	213	182
S K_{248} (0,35 kgm ²) [N m rad ⁻¹]	368	290
S K_{248} (0,097 kgm ²) [N m rad ⁻¹]	195	207
S K_{478} [N m rad ⁻¹]	575	455
S AGBRESA G [N m rad ⁻¹]	186	177
S AGBRESA H [N m rad ⁻¹]	188	187

K_{80} sowie $K_{248_{neu}}$ zeigen bei Auswertung mit der Cosinus Fit Auswertung höhere Steifigkeiten als mit der herkömmlichen Auswertung per Ableitung des Positionstreams. Alle anderen Federn sowie die ausgewählten AGBRESA Probanden weisen im Mittel niedrigere Steifigkeiten

bei der Cosinus-Fit-Auswertung auf.

Für die AGBRESA Probanden G und H wurden die Standardabweichungen ermittelt, um den Einfluss der Cosinus-Fit-Auswertung auf die Streuung der Messergebnisse zu untersuchen. Die Steifigkeit für Proband G liegt bei $(186 \pm 83) \text{ Nmrad}^{-1}$ durch Positionsableitung und bei $(177 \pm 71) \text{ Nmrad}^{-1}$ durch Cosinus-Fit. Die mittlere Steifigkeit von Proband H liegt bei einem Wert von $(188 \pm 76) \text{ Nmrad}^{-1}$ bzw. $(187 \pm 69) \text{ Nmrad}^{-1}$ für die neue Fit Methode.

4.3 Sinusoidal Perturbation

Dieses Kapitel behandelt die Ergebnisse der SP-Tests für das Phantom und für die AGBRESA-Daten. Beide Typen wurden nach der gleichen Messmethodik vermessen. Jeder Testdurchlauf des Phantoms wurde pro Permutation einmalig durchgeführt. Hieraus ergeben sich insgesamt 96 Datensätze (48 für Gain, 48 für Phase), die im Folgenden analysiert wurden. Jeder Datensatz ist eine einzigartige Kombination aus den Parametern %MVC, K, B und MVC. Weiterhin umfasst jeder Datensatz die Frequenzen von 4 bis 16 Hz. In den nachfolgenden Ergebnissen sind die Datensätze unterschiedlich gruppiert um entsprechende Unterschiede zu untersuchen.

4.3.1 Typinterne Analyse

In diesem Unterkapitel sind die Ergebnisse der typinternen Untersuchungen dokumentiert. Diese bilden die Datengrundlage zur Einschätzung des Einflusses der Input-Variablen auf das resultierende Drehmoment im Phantom. Hierbei werden die Untersuchungsvariablen entsprechend ihrer Merkmalsausprägungen gruppiert und diese Gruppen untereinander verglichen. In den nachstehenden Diagrammen sind diese Gruppen farblich unterschieden, hierbei werden die Einzelmessreihen halb-transparent dargestellt und der Mittelwert dieser Einzelmessreihen wird nicht-transparent dargestellt. In den Bode-Plots ist sowohl der Gain als auch die Phase aufgetragen, das Darstellungsprinzip ist für beide gleich. Zur quantitativen Bestimmung des Einflusses wird die über alle Frequenzen gemittelte Standardabweichung zwischen den Gruppen berechnet.

4.3.1.1 Phantom

Für die prozentuale Muskelaktivierung %MVC ergibt sich für den Gain eine Standardabweichung von $\pm 0,4$ dB. Die Phase weist eine Standardabweichung von $\pm 10^\circ$ auf.

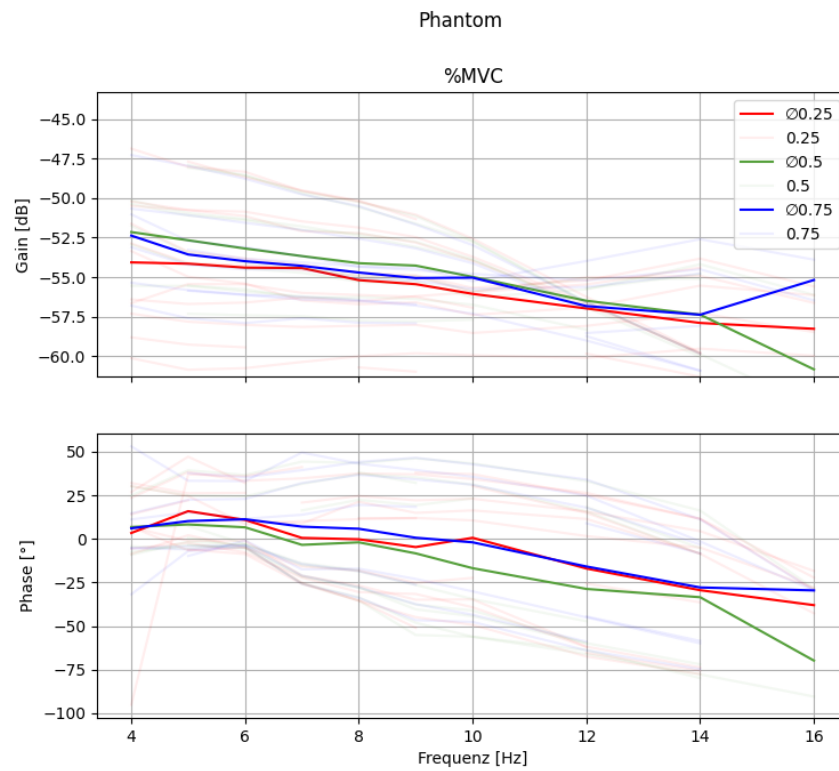


Abbildung 4.13: Farblich gruppierte Darstellung der Ausprägung des Merkmals %MVC im Phantom. Die Einzelmessreihen sind halb-transparent dargestellt, der gruppeninterne Mittelwert nicht-transparent. Rot ist hierbei der 25 %MVC-Wert, grün der 50 %MVC-Wert und blau der 75 %MVC-Wert.

Als nächstes wurde der Einfluss der MVC untersucht. Hierbei ergab sich eine Gain-Standardabweichung von $\pm 0,3 \text{ dB}$. Die Standardabweichung innerhalb der Phasengruppen beträgt $\pm 6,9^\circ$.

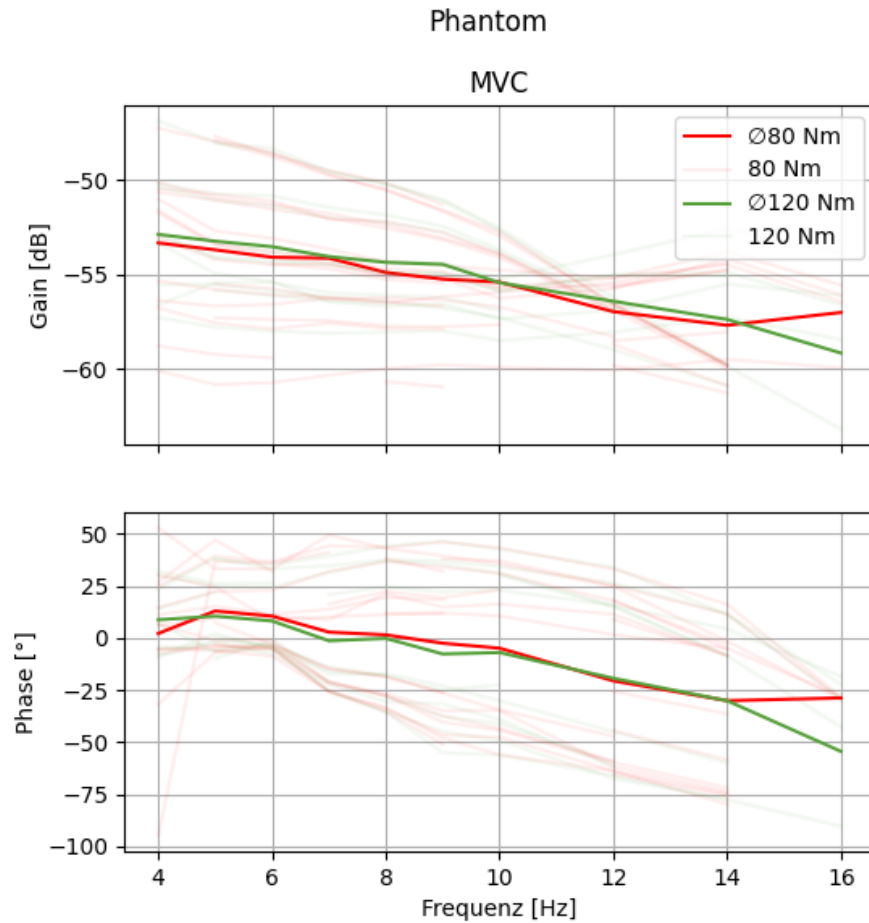


Abbildung 4.14: Farblich gruppierte Darstellung der Ausprägung des Merkmals MVC im Phantom. Die Einzelmessreihen sind halb-transparent dargestellt, der gruppeninterne Mittelwert nicht-transparent. Rot dargestellt ist MVC = 80 Nm, grün entspricht MVC = 120 Nm.

Nun wurde der Einfluss der Drehfederkonstanten K evaluiert. Es ergab sich eine Standardabweichung für den Gain von $\pm 1,4 \text{ dB}$. Die Phasen-Standardabweichung hat einen Wert von $\pm 14^\circ$.

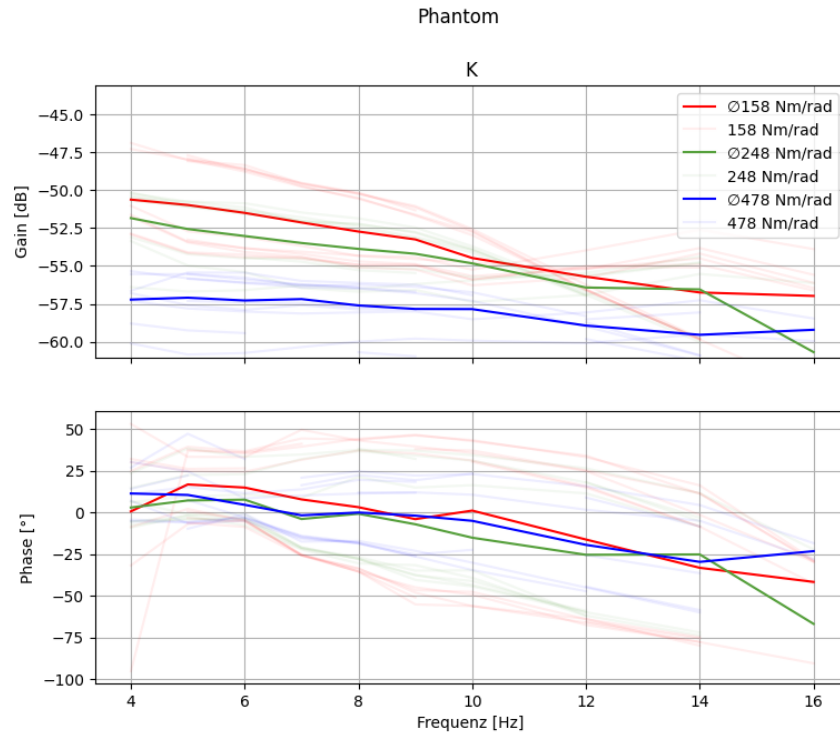


Abbildung 4.15: Farblich gruppierte Darstellung der Ausprägung des Merkmals K im Phantom. Die Einzelmessreihen sind halb-transparent dargestellt, der gruppeninterne Mittelwert nicht-transparent. Rot dargestellt sind die Werte für einen K -Wert von 158,14 Nm/rad, grün entspricht dem K -Wert 248,43 Nm/rad und blau der K -Wert für 478,08 Nm/rad.

Schließlich wurde untersucht, inwieweit die Dämpfungskonstante B das resultierende Drehmoment beeinflusst. Bezüglich des Gains beträgt die Standardabweichung $\pm 1,0 \text{ dB}$. Für die Standardabweichung der Phase ergibt sich ein Wert von $\pm 21^\circ$.

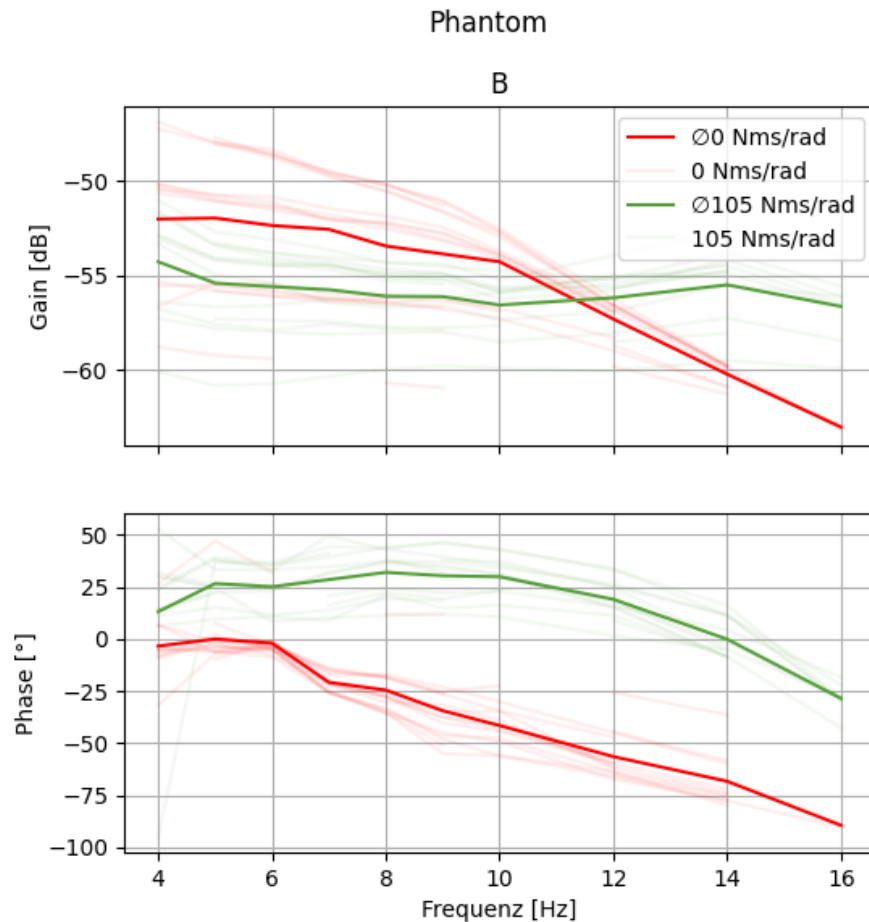


Abbildung 4.16: Farblich gruppierte Darstellung der Ausprägung des Merkmals B im Phantom. Die Einzelmessreihen sind halb-transparent dargestellt, der gruppeninterne Mittelwert nicht-transparent. Rot dargestellt sind die Messdaten für die Testung ohne Dämpfer ($B = 0 \text{ Nms/rad}$) und in grün aufgetragen sind die Daten für die Testung mit Dämpfer ($B = 105 \text{ Nms/rad}$).

4.3.1.2 Analyse der AGBRESA-Daten

Die AGBRESA-Daten wurden zu Vergleichszwecken ebenfalls analysiert. Die Messmethodik ist mit der für das Phantom identisch. Da im Falle der Erhebung der AGBRESA-Daten menschliche Probanden vermessen wurden, ist keine Aussage über den Wert der Federkonstanten K und Dämpfung B möglich. Somit lassen sich die AGBRESA-Daten lediglich an Hand von %MVC und MVC unterscheiden. Das Prinzip der Auswertung ist mit dem Prinzip für die Phantom-Auswertung identisch. Es ergibt sich im Bezug auf %MVC eine Standardabweichung des Gains von $\pm 1,1 \text{ dB}$. Die Phase weist eine Standardabweichung von $\pm 5,6^\circ$ auf.

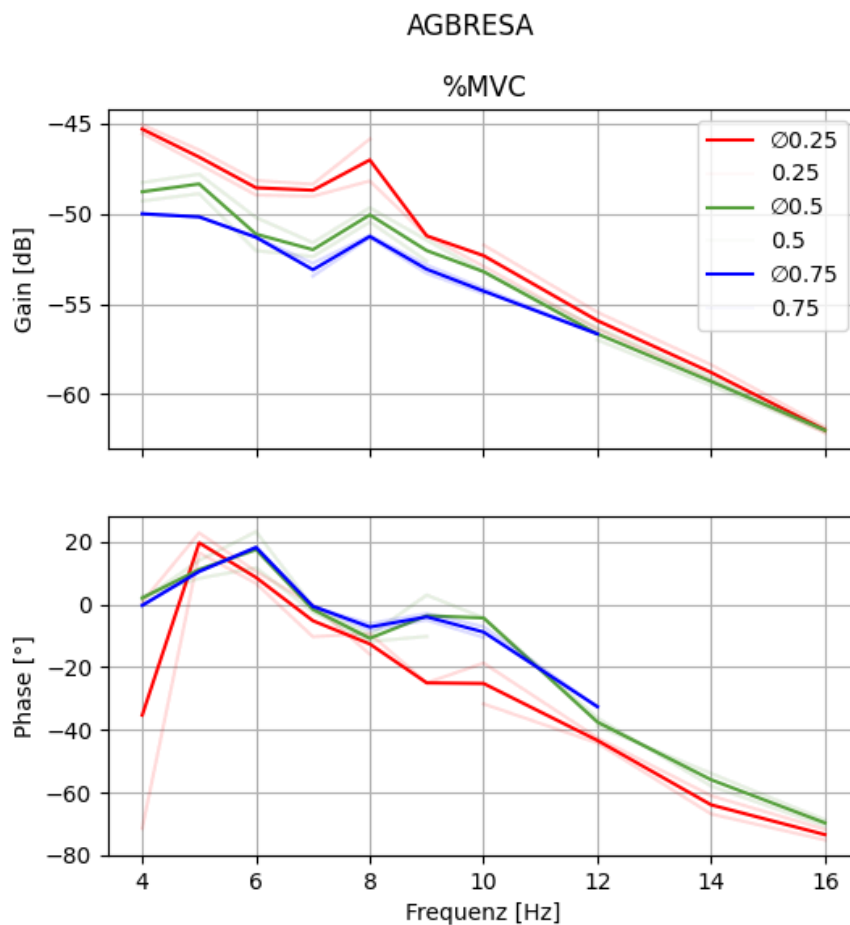


Abbildung 4.17: Farblich gruppierte Darstellung der Ausprägung des Merkmals %MVC in den AGBRESA-Messdaten. Die Einzelmessreihen sind halb-transparent dargestellt, der gruppeninterne Mittelwert nicht-transparent. Die farbliche Kodierung erfolgt wie in der Abbildung der Phantomwerte (Abb. 4.14).

Für die Standardabweichung des Gains auf Grund unterschiedlicher MVC ergibt sich ein Wert von $\pm 0,3 \text{ dB}$. Die Standardabweichung der Phase beträgt $\pm 3,0^\circ$.

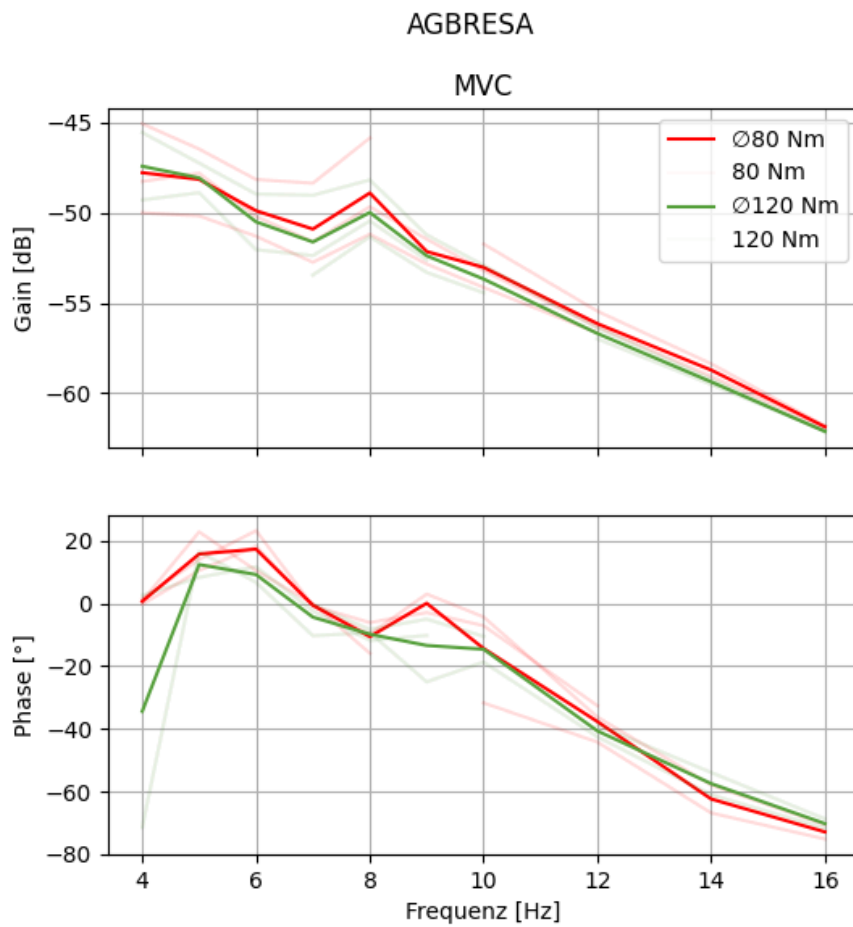


Abbildung 4.18: Farblich gruppierte Darstellung der Ausprägung des Merkmals MVC in den AGBRESA-Messdaten. Die Einzelmessreihen sind halb-transparent dargestellt, der gruppeninterne Mittelwert nicht-transparent. Die farbliche Kodierung erfolgt wie in der Abbildung der Phantomwerte (Abb. 4.14).

4.3.2 Vergleich des Phantoms mit dem mathematischen Modell

Um eine Grundlage zur Klärung der Frage, ob das mathematische Modell das Verhalten des Phantoms prognostizieren kann, zu schaffen, werden in den nachstehenden Abbildungen sämtliche Permutationen aus B und K des Phantoms mit der mathematischen Prognose verglichen. Es wird aus Übersichtsgründen darauf verzichtet, %MVC und MVC ebenfalls darzustellen, da ihr Einfluss geringer ist.

Im nachstehenden Diagramm werden die verschiedenen Drehfederkonstanten K bei einer Dämpfung $B = 0 \text{ Nmsrad}^{-1}$ dargestellt. Die Einzel-Messreihen der Phantom-Messungen sind halb-transparent in rot dargestellt. Der Mittelwert der Einzelmessreihen ist nicht-transparent aufgetragen. Das, der Permutation entsprechende, mathematische Modell ist in schwarz abgebildet. Wie im vorigen Kapitel sind die Ergebnisse als Bode-Plot dargestellt.

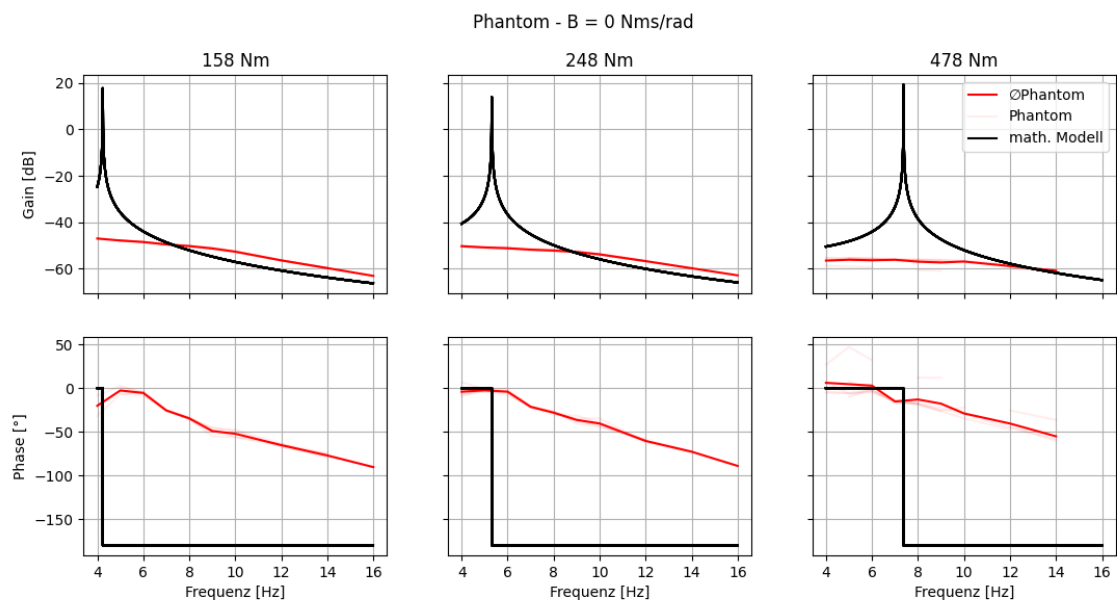


Abbildung 4.19: Vergleich der Phantom-Messdaten unterschiedlicher Drehfederkonstanten K bei Drehdämpfungskonstante $B = 0 \text{ Nmsrad}^{-1}$. Nicht-transparent rot dargestellt ist der Mittelwert der Phantomdaten, halb-transparent sind die korrespondierenden Einzelmesswerte. Das mathematisch erwartbare Ergebnis ist in schwarz aufgetragen.

Das folgende Diagramm vergleicht die Drehfederkonstanten bei Dämpfung $B = 105 \text{ Nmsrad}^{-1}$. Darüber hinaus erfolgt die Darstellung wie im vorangegangenen Diagramm.

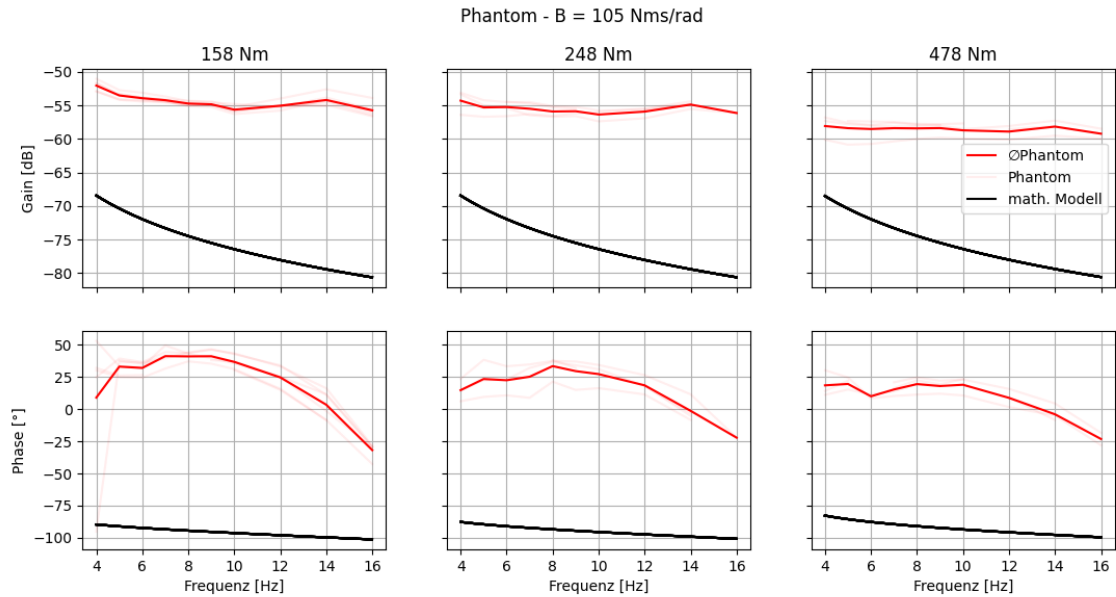


Abbildung 4.20: Vergleich der Phantom-Messdaten unterschiedlicher Drehfederkonstanten K bei Drehdämpfungskonstante $B = 105 \text{ Nmsrad}^{-1}$. Nicht-transparent rot dargestellt ist der Mittelwert der Phantomdaten, halb-transparent sind die korrespondierenden Einzelmesswerte. Das mathematisch erwartbare Ergebnis ist in schwarz aufgetragen.

Eine Näherung eines quantitativen Qualitätsmerkmals zur Überprüfung der Validität des mathematischen Modells zur Beschreibung von Gelenkdynamiken ist durch den Vergleich des Modellverhaltens bei der Frequenz 0 Hz mit den zu Frequenz 0 Hz extrapolierten Phantomkurven möglich. Diese Untersuchungen sind praktisch nicht umsetzbar, weswegen auf eine Extrapolation zurückgegriffen werden muss. Entsprechend Formel 2.1 verringert sich durch die Betrachtung der Kurve bei Frequenz 0 Hz der Einfluss der Terme I und B auf 0 , da diese Variablen frequenzabhängig sind. Hieraus folgt:

$$T(t) = K * \Theta(t) \quad (4.11)$$

Entsprechend kann mit Hilfe des Drehmoments bei Frequenz $f = 0 \text{ Hz}$ auf die Steifigkeit S des Systems zurückgeschlossen werden. Für das mathematische Modell ergeben sich folgerichtig die Federkonstanten, welche auch als Input dienen. So ergibt sich z.B. für das mathematische Modell bei $K = 248 \text{ Nm}$ an der Stelle $f = 0 \text{ Hz}$ ein Wert für $T(t)$ von 48 dB . Aus der Umrechnung von dB in Nmrad^{-1} mit $S = 10^{\frac{-48}{20}}$ folgt $S = 248 \text{ Nmrad}^{-1}$.

Selbiges Prinzip lässt sich auf die Phantomkurven übertragen. Werden diese graphisch extrapoliert, so erhält man, pro K , die Werte für $T(t)$ bei $f = 0 \text{ Hz}$. Hierbei wurde ebenfalls graphisch die Frequenz 4 bis 9 Hz als Referenz gewählt, mit Hilfe derer ein ausreichend gut vorhersagbares Verhalten zur Extrapolation erkennbar ist. Die Extrapolation ist nachstehend dargestellt.

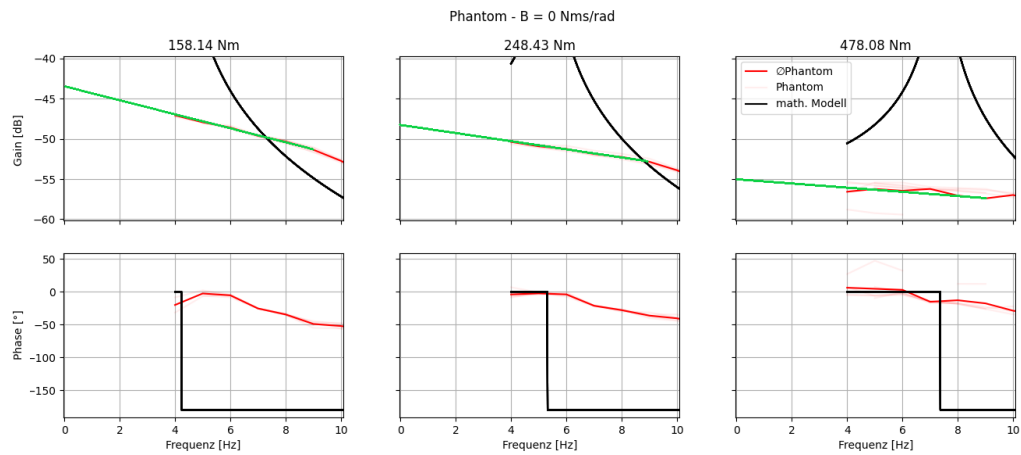


Abbildung 4.21: Graphische Extrapolation (grün) der Phantomkurven nach Frequenz $f = 0$ Hz bei Testung ohne Dämpfer ($B = 0$ Nms/rad).

Das gleiche Prinzip der Extrapolation kann auch auf die Situation bei $B = 105$ Nmsrad⁻¹ angewendet werden. Hierbei ergibt sich folgendes Bild:

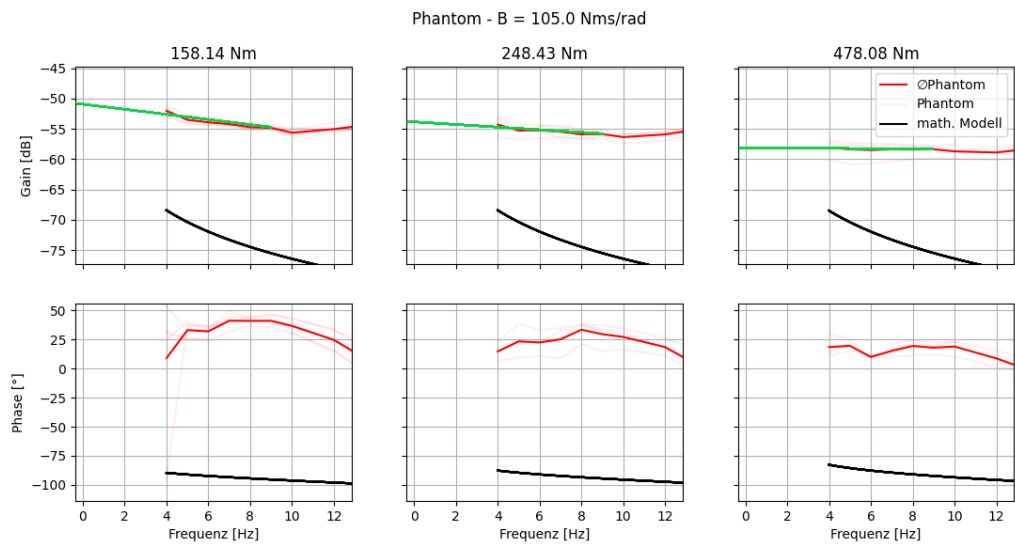


Abbildung 4.22: Graphische Extrapolation (grün) der Phantomkurven nach Frequenz $f = 0$ Hz bei Testung mit Dämpfer ($B = 105$ Nms/rad).

Weiterhin finden sich in Tabelle 4.7 die Gain-Werte des mathematischen Modells G_{Modell} und der SP-Tests mit Phantom G_{SP0} bzw. G_{SP105} bei Frequenz $f = 0$ Hz, sowie die Umrechnungen der Gain-Werte in die jeweiligen Steifigkeiten S . Schließlich ist die prozentuale Abweichung der Steifigkeiten S_{SP0} bzw. S_{SP105} zu S_{Modell} in den letzten Spalten der Tabelle eingetragen. Die Indizes der G_{SP} -Werte weisen auf die Dämpfungskonstante hin.

Tabelle 4.7: Gain-Werte G bei Frequenz 0 Hz des mathematischen Modells (G_{Modell}) und der SP-Tests des Phantoms (G_{SP0} bzw. G_{SP105}), sowie die typzugehörige Umrechnung in die Steifigkeiten K des jeweiligen Systems. Weiterhin ist die prozentuale Differenz von S_{SP0} zu S_{Modell} aufgetragen. Alle Werte zu G sind in dB angegeben. Alle S-Werte sind in Nm/rad angegeben.

	G_{Modell}	S_{Modell}	G_{SP0}	S_{SP0}	G_{SP105}	S_{SP105}	$\Delta S_{SP0} S_{Modell} [\%]$	$\Delta S_{SP105} S_{Modell} [\%]$
K_{158}	44	158	44	150	51	355	-5	125
K_{248}	48	248	48	251	54	501	1	102
K_{478}	54	478	55	562	58	794	18	66

Wie in der Spalte $\Delta S_{SP0} S_{Modell}$ der Tabelle 4.7 zu sehen, liegen die mathematisch zu erwartenden und experimentell bestimmten Werte der Steifigkeiten S bei Frequenz 0 Hz und ohne Dämpfer ($B = 0 \text{ Nmsrad}^{-1}$) für $K = 158 \text{ Nmrad}^{-1}$ bzw 248 Nmrad^{-1} sehr nah beieinander (-5 bzw. 1 %). Die S-Werte für $K = 478 \text{ Nmrad}^{-1}$ weichen jedoch etwas stärker ab (18 %). Betrachtet man die Steifigkeitswerte mit Dämpfer ($B = 105 \text{ Nmsrad}^{-1}$), so zeigen sich erheblich größere Abweichungen von dem mathematisch zu erwartenden Wert. Hierbei weicht der Wert für K_{158} um 125 % ab, der Wert für K_{248} um 102 % und schließlich der Wert für K_{478} um 66 %.

Kapitel 5

Diskussion

Im Folgenden werden die Ergebnisse der Arbeit bezüglich der Konstruktion des Phantoms sowie der allgemeinen Testmethodik, Testreihen und Messgeräte und Durchführung diskutiert.

5.1 Validierung der grundlegenden Parameter

Die nachfolgenden Unterkapitel befassen sich mit den Ergebnissen der Inputvariablen, welche die Grundlage für die Berechnung des mathematisch zu erwartenden Verhaltens bilden.

5.1.1 Überprüfung des Trägheitsmoments

Das Trägheitsmoment wurde gemäß der Formeln aus Kapitel 3.4 experimentell zu einem Wert von $0,223 \text{ kgm}^2$ bestimmt. Die Plausibilität dieses Wertes wurde durch eine theoretische Berechnung des Trägheitsmomentes überprüft. Unter Annahme eines Quaders als Leverarm mit den gemessenen Dimensionen $0,043 \text{ m} \times 0,12 \text{ m} \times 0,34 \text{ m}$ bei einer gemessenen Masse von $3,45 \text{ kg}$ ergibt sich ein Teilträgheitsmoment von $0,034 \text{ kgm}^2$. Hinzu kommt noch eine Verschiebung des Massenschwerpunktes im Bezug auf die Drehachse, wodurch sich ein verschobenenes Massenträgheitsmoment von $0,083 \text{ kgm}^2$ ergibt. Schließlich wird das Phantom inklusive Adapter als Punktmasse am Ende des Quaders berücksichtigt. Bei einer Masse von $3,296 \text{ kg}$ und einem Hebelarm von $0,238 \text{ m}$, ergibt sich hier ein weiteres Teilträgheitsmoment von $0,187 \text{ kgm}^2$. Aufgrund der additiven Natur von Trägheitsmomenten ergibt sich für das errechnete Gesamt-massenträgheitsmoment ein Wert von $0,270 \text{ kgm}^2$. Dieser Wert weicht um ca. 17 % von dem experimentell bestimmten ab, was im Rahmen dieser Proof-of-Concept-Untersuchungen zur Bestätigung der richtigen Größenordnung unseres errechneten Massenträgheitsmomentes genügt. Für zukünftige Untersuchungen dieser Art, sollte allerdings die Genauigkeit und Verlässlichkeit der Bestimmung des Massenträgheitsmomentes erhöht werden. Hierbei wäre es denkbar, eine simplere Konstruktion zu wählen, um das Massenträgheitsmoment mathematisch exakt bestimmen zu können oder die Messmethode zu professionalisieren, um zuverlässigere Ergebnisse zu erhalten.

5.1.2 Abweichungen der Dämpfungskonstanten

Wie auch die Genauigkeit und Verlässlichkeit der Werte des Trägheitsmomentes, sind die Methoden zur Bestimmung der Dämpfungskonstanten für die Durchführung unserer Testreihen ausreichend, sollten allerdings für zukünftige, genauere Untersuchungen präziser bestimmt werden. Alternativ sollte ein nicht-verstellbarer Dämpfer genutzt werden, dessen Dämpfungskonstante mit sehr hoher Verlässlichkeit unveränderlich ist. Von großer Relevanz ist auch der Wert der Dämpfungskonstanten, dieser wird in einem späteren Kapitel gesondert diskutiert.

5.1.3 Abweichung der ermittelten Federkonstanten von den Herstellerangaben

Die in Kapitel 4.1.3 ermittelten Federkonstanten bewegen sich im Rahmen eines Fehlers von ca. 10 % zu den Herstellerangaben. Grundsätzliche Abweichungen liegen vermutlich in der Ungenauigkeit der Testmethodik begründet. Im Falle der Feder 2,89 N/mm sind erhebliche Abweichungen von den Herstellerangaben festzustellen (26 %). Über die Abweichungen des Wertes der Federkonstanten hinaus, liegt bei dieser Feder auch ein sehr viel geringeres Bestimmtheitsmaß ($R^2 = 0,8902$) vor. Das Bestimmtheitsmaß ist ein Maß für die Linearität der Feder, welche in allen anderen Federn einen Wert von 0,9999 oder höher hat. Die Linearität aller Federn außer der Feder mit einer Federkonstanten von 2,89 N/mm ist als bestätigt zu sehen. Eine Wiederholung der Messreihen für die Feder war aus Zeitgründen nicht möglich. Auf Grund der extremen und außergewöhnlichen Abweichungen dieser Feder, und der sehr hohen Linearität der anderen Federn, ist davon auszugehen, dass durch eine weitere Testreihe auch die Linearität dieser Feder validiert werden könnte. Für die weitere Diskussion wird angenommen, dass ein Fehler in der Durchführung zu den Abweichungen geführt hat und nicht, dass die Feder tatsächlich in so großem Maße abweicht. Die Sicherstellung der Linearität der Federn war essentiell, um die Zuverlässigkeit der Ergebnisse sicherzustellen. Grund zum Zweifel an der Linearität wurden durch QR-Ergebnisse hervorgerufen, die eine Nicht-Linearität der Federn suggerierte. Dieser Verdacht konnte nicht bestätigt werden.

5.2 SPQR-Phantom

Das SPQR-Phantom bildet die Grundlage für die Validierung der Messmethoden und muss entsprechend gewissen Ansprüchen genügen. Das Lastenheft (Abschnitt 3.3.1) charakterisiert dabei diese Ansprüche und teilt sie in Musskriterien, die das Phantom zum Erreichen der gewünschten Ziele zwingend erfüllen muss sowie Sollkriterien die nicht zwingend erforderlich sind, die Funktionalität des Phantoms jedoch verbessern und ggf. auch in zukünftigen Studien, welche das Phantom nutzen, hilfreich sein können. Hierbei ist von größter Wichtigkeit, dass mit Hilfe des Phantoms das zu ermittelnde Merkmal

- eindeutig festgestellt werden kann, die Ergebnisse also von der Person des Untersuchers unabhängig sind (Objektivität).

- exakt erfasst werden kann, bei wiederholter Anwendung unter gleichen Bedingungen sich das Ergebnis also nicht ändert (Reliabilität).
- zweifelsfrei mit dem angewandten Verfahren tatsächlich gemessen werden kann (Validität).

Im anschließenden Kapitel wird auf die erfolgreichen Aspekte der Konstruktion des Phantoms, sowie auf Unzulänglichkeiten und das Verbesserungspotential desselbigen eingegangen. Abschließend werden nicht-strukturelle Probleme beleuchtet.

5.2.1 Erfolgreiche Aspekte des Designs

Den Muss-Kriterien des Lastenhefts (Abschnitt 3.3.1) konnte mit dem gefertigten Phantom entsprochen werden. Hierzu zählt der Verbau einer viskosen und einer elastischen Komponente (in Form eines Dämpfers und einer Feder), die generelle Funktionalität bei QR und SP, durch die Austauschbarkeit von Feder und Dämpfer die Möglichkeit verschiedene Steifigkeiten zu modellieren, die Möglichkeit verschiedene Kraftabstufungen zu simulieren (durch Aufziehen der Federn mit Hilfe von Augenschrauben in Gewindebohrungen und die Erzeugung realistscher Drehmomente. Die Drehmomente erreichten entsprechend unseres Testdesigns nicht die Werte von humanen Probanden, das Phantom ist jedoch generell geeignet mit stärkeren Federn zu funktionieren. Ein Erreichen und sogar Übersteigen der Maximaldrehmomente von humanen Probanden ist also umsetzbar.

Die Soll-Kriterien konnten nur teilweise erfüllt werden. Die Vorspannung der Federn kann durch die Augenschrauben während der Testreihen variiert werden und es kann schnell auf die wechselnden Vorspannungen bei unterschiedlichen %MVC reagiert werden. Das optimale Trägheitsmoment des Phantoms wurde im Lastenheft als möglichst klein angegeben, im Verlauf des Konstruktionsprozesses hat sich jedoch gezeigt, dass das optimale Trägheitsmoment nahe des Trägheitsmomentes der Konfiguration für humane Probanden liegt wie es bei AGBRESA genutzt wurde. Dadurch sind die Daten der Versuche weitaus vergleichbarer. Das Phantom liegt mit einem Trägheitsmoment von $0,223 \text{ kgm}^2$ nicht genau bei dem Wert der Fußhalterung der AGBRESA Studie von $0,35 \text{ kgm}^2$. Für eine gewisse Vergleichbarkeit sollte der Wert jedoch genügen. Schlussendlich ist die Möglichkeit verschiedene Trägheiten zu modellieren das letzte Soll-Kriterium. Dieses Kriterium wurde als einziges nicht bei der Konstruktion des Phantoms umgesetzt. Gleichzeitig ist jedoch festzuhalten, dass das Massenträgheitsmoment durch das Zuführen von Massen in gewissen Abständen zur Drehachse variiert werden kann. Folglich ist eine Messdurchführung zumindest mit erhöhten Massenträgheitsmomenten zukünftig möglich, obwohl dies nicht explizit als Funktion des Phantoms implementiert wurde.

Ein weiterer erfolgreicher Aspekte des Phantoms ist die Befestigung an den Fixing-Adaptern des MARES, welche wie geplant funktioniert hat. Die Fixing-Adapter wurden ausgemessen und anhand dieser Werte das Phantom konstruiert. Ein Messfehler hätte hier zur kompletten Neuanfertigung des Designs geführt. Außerdem konnte während der Tests festgestellt werden, dass die Steifigkeit des Phantoms ausreichend ist, um zuverlässig und über einen langen Zeitraum

Ergebnisse zu reproduzieren. Auch die Idee, das Federn-Dämpfer-System mit Hilfe von einschraubbaren Augenschrauben in Kombination mit Schäkeln zu befestigen, hat in weiten Teilen zu zufriedenstellenden Ergebnissen geführt, wobei jedoch auch einige Unzulänglichkeiten auftraten, welche im anschließenden Kapitel behandelt werden.

5.2.2 Verbesserungspotential des Phantoms

Das erste Problem welches im Zusammenhang mit dem Phantom auftrat, war ein rein technisches Konstruktionsproblem. Die Bohrungen und Gewinde, welche als Aufnahme für die Augbolzen bzw. Augenschrauben dienten, wurden von der fertigenden Firma als zwei separate Gewindebohrungen behandelt. Somit konnten die Schrauben nicht wie gewollt in beide Gewinde eingeschraubt werden. Dieser Missstand wurde mit Hilfe der DLR-internen Werkstatt behoben, indem eine Reihe der Gewinde ausgebohrt wurde. Nach eigener Einschätzung hat dies keine abträglichen Effekte auf die Testdurchführung zur Folge.

Zu Beginn der ersten Testdurchläufe wurde weiterhin festgestellt, dass die Art der Aufhängung einige Probleme mit sich brachte. So wurde in ersten Prototypen des Designs die Befestigung per Augbolzen und nicht per Augenschrauben durchgeführt. Dies führte dazu, dass kein bündiges Interface vom MARES zum Feder-Dämpfer-System herzustellen war. Eine Befestigung mit Hilfe von Muttern, Schrauben und Unterlegscheiben führte zu zahlreichen, potentiell störenden Schnittstellen. In Folge dessen wurde das Augbolzen-System durch eine Konstruktion mit Augenschrauben und Schäkeln ersetzt, welche durch normierte M8-Aufnahmen, formschlüssig mit dem Feder-Dämpfer-System gekoppelt werden konnten. Hierdurch wurden eine Reihe ungewisser Störeinflüsse eliminiert. Es gibt jedoch ein grundlegendes Problem der Befestigung mit Hilfe von Schäkeln. Grundlegend ist eine komplett steife Befestigung bei den Federn im Gegensatz zum Dämpfer nicht nötig, da diese durchgehend auf Zug belastet sind. Das gilt sowohl für SP bei denen die Feder dauerhaft gespannt ist, als auch für QR, bei dem die Feder zwar so weit zurückspringt, dass sie nicht mehr gespannt ist, der relevante Messbereich jedoch lediglich die ersten Millisekunden umfasst in dem die Feder noch auf Zug ist. Für sehr kleine Drehmomente (primär 10 Nm) hat sich bei den QR-Messungen jedoch gezeigt, dass die Feder den linken Teil des Phantoms so gering beschleunigt, dass der Schäkel zum Teil die Zeit hat, zur Seite auszuweichen. Dies beeinflusst die Werte insofern, dass nicht mehr die reine Feder vermessen werden kann. Um dieses Problem zu beseitigen, wäre eine Befestigung von Nöten, die eine geringere Anzahl an Freiheitsgraden umfasst.

Ein weiteres Problem des Phantoms bei kleinen Drehmomenten, ist die geringe Auslenkung der Federn. Dies bezieht sich primär auf Federn mit hohen Federkonstanten. Federn mit hohen Federkonstanten in Kombination mit geringen Drehmomenten führen dazu, dass die Federn nur sehr gering ausgelenkt werden, um diese Drehmomente zu erreichen. Das führt besonders bei Quick Release Messungen zu einer drastischen Verkleinerung des möglichen Auswertungsintervalls, da die Feder sehr früh wieder ihre Ausgangsauslenkung erreicht. Für Sinusoidal Perturbation Messungen ergibt sich dieses Problem nicht, da die Auslenkung mit 3 ° immer gleich und damit i.d.R.

kleiner als die kleinste Auslenkung der Feder ist. Ein Aufbau mit geringen Federkonstanten in Kombination mit dafür höheren Federauslenkungen ist dementsprechend künftig zu präferieren. Weiterhin ergeben sich im Bezug auf die Federn und den Dämpfer viele Möglichkeiten zur Optimierung. Während die Federn noch gut nutzbar waren, gestalten sich die Eigenschaften des Dämpfers als problematischer. Die Auswirkungen der Dämpfungskonstante waren während des Konstruktionsprozesses schwer abschätzbar, weshalb ein stufenlos verstellbarer Dämpfer bestellt wurde, welcher sich nach der Datenakquise, selbst in leichtgängigster Einstellung, als tendenziell zu stark dämpfend herausstellte. Das Bestellen und die Nutzung eines geeigneteren Dämpfers konnte aus Gründen der Verfügbarkeit und durch zeitliche Einschränkungen nicht realisiert werden. Hier könnte, unter Verwendung eines geeigneteren Dämpfers, ein größerer Datenbereich abgedeckt werden. Auch ist zu überlegen, durch das Einfügen einer Scherenkonstruktion, den vorliegenden Dämpfer besser nutzbar zu machen. Siehe hierzu auch Kapitel 5.1.2

Ein letzter Punkt ist die Arbeitssicherheit bei Benutzung des Phantoms. Während der durchgeführten Versuche, befanden sich die Testleiter durchgehend hinter einer Trennwand. Das Verstellen der Vorspannung geschieht bei QR zwischen den einzelnen Messungen unterschiedlicher %MVC. Hier wird die Feder entweder weiter hinein oder heraus geschraubt um die Vorspannung zu variieren. Der MARES-Arm ist zu diesem Zeitpunkt nicht in Nullposition und damit sind die Federn nicht gespannt. Das Verstellen ist damit zwar grundlegend ungefährlich, theoretisch würde eine zu langsame Verstellen der Vorspannung, beispielsweise durch einen unerfahrenen Nutzer, dazu führen, dass der Testleiter zum Zeitpunkt des Spanns der Federn noch in der Nähe des Versuchsaufbau ist. Diese Problematik ist bei den vorliegenden Federkräften ein kleineres Problem, sollte aber in zukünftigen Testreihen bei potenziell höheren Federkräften berücksichtigt werden. So wäre es z.B. wünschenswert, einen Mechanismus zu erdenken, der die Einstellung der Federspannung aus der Ferne ermöglicht, bzw. so, dass zu keinem Zeitpunkt die Möglichkeit einer Gefährdung besteht. In einem erweiterten Schritt, wäre eine Automatisierung der Federeinstellung denkbar, wodurch ein Einfluss auf die Ergebnisse durch die händische Einstellung minimiert werden könnte.

5.3 MARES

Im Zuge der Anfertigung dieser Masterarbeit wurden zahlreiche Fragen bezüglich der Funktionsweise des MARES aufgeworfen. Hieraus entsprang die Erkenntnis, dass das Trägheitsmoment, welches das MARES ausgleicht, eingestellt werden kann. Auf Grund dieser Erkenntnis wurden beispielhaft Messungen mit veränderter Trägheitsmomenteinstellung durchgeführt. Es zeigte sich jedoch im Bezug auf die SP-Messungen keine Veränderung der Messergebnisse, weshalb von einer Wiederholung der SP-Messreihen unter veränderten Bedingungen abgesehen wurde. Der Einfluss auf die QR-Messungen wird im Rahmen der entsprechenden Ergebnisse und Diskussion behandelt.

Bezugnehmend auf die Problematik der größtenteils unbekannten MARES-Software, ergab sich

ein Störeinfluss, welcher jedoch aus dem Weg geräumt werden konnte. Hierbei ging es um das Profil, welches im Normalfall die Rahmenparameter des Probanden erfasst und abspeichert, um eine gefahrungsfreie Durchführung der Tests sicherzustellen. Im Rahmen der Charakterisierung des Modellverhaltens während der Tests wurde bei der Auswertung der Daten festgestellt, dass die Endstops, welche die Range of Motion (RoM) festlegen, zu wenig Spielraum für den vollen Bewegungsumfang boten. Dies wirkte sich insofern negativ auf die Tests aus, dass zum Schutz der mechanischen Elemente des MARES ein softwareseitiger Eingriff stattfand, welcher den geplanten Testablauf veränderte. Dieses Problem konnte durch eine veränderte Einstellung des Profils beseitigt werden.

Im Zusammenhang mit dem im vorangegangenen Kapitel angesprochenen Problem der Kopplung von Feder-Dämpfer-System und MARES trat ein weiteres Problem auf. Im gekoppelten Zustand konnten willkürlich einsetzende Vibrationen des Feder-Dämpfer-Systems beobachtet werden. Aufgrund der Kopplung stand jedes Teil des Systems als Ursache im Verdacht. Mit Hilfe einiger klärender Tests konnten diese Ursachen genauer bestimmt werden. Hieraus begründete sich unter anderem auch der Wechsel der Art der Befestigung des Systems, um die Anzahl der Schnittstellen als potentielle Fehlerquellen zu minimieren. Schließlich konnte das Problem so sehr eingegrenzt werden, dass ein softwareseitiger Eingriff des MARES im Sinne einer Ausgleichs- bzw. Schutzantwort als wahrscheinlichste Ursache identifiziert wurde. Dieses Erkenntnis wurde durch die Kommunikation mit einem Softwareingenieur des MARES-Projektes konsolidiert. Die Begründung hierfür ist ein sehr hoher Anspruch an die Sicherheit des Systems zum Schutze der humanen Testpersonen, weshalb das MARES auf Inputs und Outputs die außerhalb einer MARES-internen Toleranz liegen, sehr sensibel reagiert. Da die Methodik zur Untersuchung unseres Phantoms und der entsprechende Aufbau von der Versuchssituation mit einem menschlichen Probanden abweicht, ist es plausibel, dass das MARES interne Regelungen vornimmt, um z.B. Resonanzkatastrophen oder ähnliches zu verhindern, welche sonst zu Schäden an Mensch oder Maschine führen würden. Da ein Eingriff in die MARES-Software im Rahmen dieser Arbeit in vielerlei Hinsicht unverhältnismäßig gewesen wäre, wurden die Faktoren, welche Vibrationen begünstigten, so weit es ging eliminiert bzw. minimiert, um so einen möglichst reibungslosen und zuverlässigen Testablauf zu gewährleisten.

Eine weitere Problematik die zu erheblichen Zeitverlusten geführt hat, war der Verlust von Daten. Das MARES hat hierbei unregelmäßig nach ca. 2 1/2 Stunden Aufnahmezeit keine weiteren Daten gespeichert, wodurch vereinzelt Datensätze unvollständig waren. Auf Grund der Masse der Daten und der Unregelmäßigkeit des Auftretens dieses Problems, fiel dieser Missetand erst nach langer Zeit auf, woraufhin alle Messreihen analysiert und entsprechend unvollständige Messreihen wiederholt werden mussten. Für diese Wiederholungen wurde das MARES ca. alle zwei Stunden neu gestartet, um einen erneuten Datenverlust auszuschließen.

Weiterhin gab es erhebliche Probleme durch Abstürze des MARES. Scheinbar willkürlich schaltete sich das MARES ab und ließ sich für einige Minuten nicht neu hochfahren. In Folge dessen gingen die Datensätze der aktuellen Sessions verloren und mussten später wiederholt werden.

Nach einigen Wochen konnte eine Überhitzung des MARES, in Folge der ausgefallenen Klimatisierung des Raumes, in dem das MARES stand, als wahrscheinlichste Fehlerursache ausgemacht werden. Hierauf wurde reagiert, indem sowohl die periphere Stromversorgung als auch das MARES selbst in Zukunft mit Ventilatoren gekühlt wurde.

Eine weitere zeitraubende Situation ergab sich auf Grund einer fehlerhaften Verarbeitung der Rohdaten durch das MARES. Hierbei gab es sowohl den Fehler, dass das MARES einen Teil der Datenstreams am Anfang oder am Ende beschnitt, sowie den Fehler, dass das MARES die Rohdaten nicht richtig verpackte, woraufhin sie nicht dekodiert werden konnten und die so gesammelten Daten verloren gingen. Ob das MARES einen dieser Fehler machte, konnte immer erst am Ende einer Messsession untersucht werden, wodurch zum Teil ganze Messtage ergebnislos blieben.

5.4 Testreihen

Zuvorderst muss zur Kenntnis genommen werden, dass bei der durchgeführten Testmethodik, unabhängig von SP und QR, die Problematik auftritt, dass neben den Testverfahren immer auch das konstruierte Phantom, das MARES selbst, sowie die Auswertungsmethodik mitbeurteilt werden. Aussagen über die Validität der Messmethodik QR und SP lassen sich nur mit vollständiger Sicherheit treffen, wenn das Phantom, das MARES sowie die Auswertung der Daten mittels der Python Skripte vollständig reliabel, valide und objektiv wären. Eine Beurteilung des SPQR-Phantoms und des MARES bezüglich dieser Kriterien findet sich in den vorangegangenen Kapiteln. Die Auswertungsmethodik mit Hilfe von Python wurde eigens für dieses Projekt entwickelt. Sie repräsentiert grundlegend die Auswertung von QR- und SP-Daten wie sie in der Literatur durchgeführt wird. Die Beurteilung der Validität des Quick Release basiert also zum Teil auf der Annahme, dass sowohl das SPQR-Phantom, das MARES sowie die Auswertungsmethodik die Gütekriterien für Modelle und Testverfahren erfüllen. Um hier eine größere Zuverlässigkeit der Daten zu erreichen, wäre eine Ausweitung der Testreihen und damit eine Erhöhung der Messwerte zielführend.

Ein weiterer zu beachtender Punkt ist das händische Einstellen der Maximal Voluntary Contraction (MVC) zu Beginn aller Testreihen. Die Federn wurden zu diesem Zwecke temporär ausgehängt und der MVC per Druck der Testleiter gegen den short lever eingestellt. Theoretisch wäre ein Einstellen des MVC durch eine eingehängte Feder genauer auf den gewünschten Wert möglich, da die Feder in den drei Wiederholungen jeweils das gleiche Moment erzeugen würde. In der Theorie ist dieser Ansatz zu bevorzugen. Die Praxis hat jedoch zwei Problematiken der Einstellung des MVC per Federkraft aufgezeigt. Das MARES fährt im Laufe der Angle Torque Relation zum Einstellen des MVCs verschiedene Winkel ab. Bei einigen dieser Winkeleinstellungen würden die Federn über ihre maximale Zugweite hinaus aufgezogen werden, was diese beschädigen würde und ein Sicherheitsrisiko darstellt. Weiterhin zeigten erste Versuche in die Richtung, dass starke Schwingungen auftreten, wenn bei der Angle Torque Relation dauerhaft

eine Kraft auf den Arm wirkt und nicht nur zum Zeitpunkt, zu dem das MARES die Kraft zur MVC-Bestimmung fordert. Diese Problematik könnte durch einen Mechanismus zum Lösen der Vorspannung der Feder mittels eines Hebels gelöst werden.

5.5 Quick Release

Das Ziel der durchgeführten Messreihen ist es, die Validität des Quick Release Tests zur Steifigkeitsbestimmung der viskoelastischen Komponente der Muskeln zu bestimmen und Schwachstellen dieser Messmethodik aufzudecken. Dazu wurde ein Phantom der viskoelastischen Eigenschaften des *M.triceps surae* konstruiert und an diesem Quick Release Tests durchgeführt, deren Ergebnisse bezüglich der Steifigkeit mit den eingestellten Drehfederkonstanten des Systems abgeglichen werden können.

Im Folgenden werden die in Abschnitt 4.2 behandelten Ergebnisse diskutiert. Zu Quick Release Messungen an einem Viskoelastizitäts-Phantom gibt es keine etablierte Vergleichsliteratur. Daher können generierte Daten lediglich in die Gesamtheit der Literatur zu Quick Release Versuchen an menschlichen Probanden eingeordnet und mit dieser verglichen werden.

5.5.1 Auswertungsmethodik des Quick Release

Zur Steifigkeitsbestimmung der verschiedenen Federkonfigurationen des Phantoms wurde eine angepasste Auswertungsmethodik nach Lambertz et al. [13] benutzt und in ein Python Skript zur Automatisierung übertragen.

Die grundlegende Auswertungsmethodik entspricht dabei dem Standard für Quick Release Messungen. Wie gut diese Methodik durch die Programmierung in Python umgesetzt wurde, ist schwierig zu evaluieren, da die Ergebnisse von mehr als nur dem Parameter der Auswertung abhängen. Es besteht eine Latenzzeit zwischen dem Stream der Geschwindigkeit und Beschleunigung im MARES (vgl. Abbildung 3.8). Um zu umgehen diese Latenzzeit schätzen zu müssen, wurde der Positionstream zwei Mal abgeleitet um die Beschleunigung zu erhalten, anstatt die Geschwindigkeit einmal abzuleiten, was die Alternative dargestellt hätte.

Eine generelle Problematik der automatisierten Auswertung ist die Erkennung der QR Intervalle. Die Intervalle werden zwar anhand der "Time Marks" verlässlich richtig erkannt, anhand der Anzahl der QRs ist jedoch abzulesen, dass nicht alle einzelnen QRs überhaupt erkannt werden. Die Validität der Auswertungsmethodik wurde zu diesem Zeitpunkt nicht validiert, da sie eigens für die vorliegende Arbeit angefertigt wurde. Da in der vorliegenden Arbeit auch grundlegende Probleme mit dem MARES festgestellt wurden, kann diese Auswertungsmethodik im Rahmen dieser Arbeit nicht vollständig validiert werden. In zukünftigen Messreihen mit dem SPQR-Phantom und der entsprechenden Auswertungsmethodik sollte diese Validierung z.B. anhand eines einfachen Dynamometers mit weniger potenziellen Fehlerquellen vorgenommen werden.

5.5.2 Bestimmung der Federsteifigkeit K

Es wurde das angelegte Drehmoment bei den Quick Release Messungen mit der resultierenden Steifigkeit in Relation gesetzt. Die Ergebnisse der Steifigkeitsmessungen der vier gemessenen Federn in Abbildung 4.2 zeigen eine Abhängigkeit der Steifigkeit vom im Versuch angelegten Drehmoment. Hierbei weisen die Federn der beiden höheren Drehfederkonstanten K_{248} sowie K_{478} eine positive Korrelation zwischen Drehmoment und Steifigkeit auf. Bei einer definierten negativen Richtung der Auslenkung im QR-Versuch, bedeutet das, dass für niedrige Drehmomente die Steifigkeiten besonders hoch sind. Die Steifigkeiten der Federn mit den zwei niedrigen Drehfederkonstanten K_{80} und K_{158} zeigen bis auf die letzten beiden Werte der Feder K_{158} eine negative Korrelation mit dem angelegten Drehmoment. Bei kleinem Drehmoment tritt hier also eine geringe Muskelsteifigkeit auf.

In der Literatur wurde ein linearer Zusammenhang der Muskulotendinösen Steifigkeit und dem vom Proband ausgeübten Drehmoment festgestellt, bei dem die Steifigkeit der SEC mit steigendem ausgeübten Moment steigt [13][28][29][30][79]. Die beiden starken Federn K_{248} und K_{478} bilden diese Entwicklung nicht ab. Hier wurden für geringe Drehmomente hohe Steifigkeiten ermittelt, es liegt also eine umgekehrte Tendenz vor. Die Steifigkeiten der Feder K_{158} bilden im Bereich zwischen 20 und 80 Nm die Entwicklung der Literatur ab, die beiden hohen Messwerte im Niedrigbereich des Drehmomentes fallen wiederum stark aus der Reihe. Die schwächste Feder K_{80} zeigte keine signifikante Korrelation mit dem Drehmoment. Es wurden weniger Messwerte als mit den anderen Federn aufgenommen, da sie im Vergleich zu den anderen Federn eine starke Vibration im Laufe der Versuche zeigte und einige Messwerte in Folge nicht ausgewertet werden konnten.

Insgesamt konnte der Zusammenhang zwischen Drehmoment und Steifigkeit der elastischen Komponente des Muskels in der Literatur bis auf einen Drehmomentsbereich von 20 – 80 Nm der Feder K_{158} nicht repliziert werden. Dabei ist festzuhalten, dass Steifigkeiten bei niedrigen Drehmomenten eine höhere Intervall-Standardabweichung aufweisen (Abbildung 4.4). Insgesamt wurden die Steifigkeiten hier also aus weiter auseinander liegenden Steifigkeitswerten gemittelt was auf eine starke Schwankung der Winkelposition des SLA innerhalb der einzelnen QRs schließen lässt. Die Messwerte sind entsprechend weniger verlässlich als Messwerte höherer Drehmomente, was wiederum ein Grund für die Abweichung zu der Steifigkeitsentwicklung über verschiedene Drehmomente in der Literatur sein kann. Die Qualitätsgrenze der Intervall-Standardabweichung weiter herunter zu setzen, um diese Problematik zu vermeiden, ist mit vorliegenden Datensätzen nicht möglich, da zu wenige QR Messwerte vorhanden sind, um verlässliche Ergebnisse zu erzeugen. Um zu untersuchen, ob das Phantom die Entwicklung der Literatur abbilden kann, sollten weitere QR-Testreihen in höheren Drehmomentsbereichen wie bei menschlichen Probanden durchgeführt werden und eine höhere Anzahl an Messwerten generiert werden.

Die Ergebnisse zur Abweichung der Federsteifigkeit zur Drehfederkonstante zeigen, dass generell sehr hohe Abweichungen vom Erwartungswert gemessen werden. Die durchschnittliche Abwei-

chung der ermittelten Steifigkeit zur Drehfederkonstante liegt über alle Federn gemittelt bei 39%. Dieser Wert wird jedoch stark von den hohen Abweichungen bei kleinen angelegten Drehmomenten beeinflusst. Die niedrigsten Abweichungen wurden durchschnittlich zwischen 30 und 40 Nm gemessen und liegen bei 23 bzw 21 %.

Die am nächsten am Erwartungswert liegende durchschnittliche Steifigkeit einer einzelnen Feder wurde mit 33% für K_{248n} ermittelt. Hierbei muss berücksichtigt werden, dass die Ergebnisse dieser Messung im Vergleich zum alten Trägheitsmoment der gleichen Feder nur im 20 Nm Bereich näher am Erwartungswert liegen. In hohen Drehmomentsbereichen schneidet diese Feder schlechter ab. Der Durchschnitt spiegelt dies jedoch nicht wider, da K_{248} bei 30 Nm mit 79% sehr stark abweicht.

Die vorliegenden Daten implizieren eine höhere Verlässlichkeit der Messmethodik in Drehmomentsbereichen zwischen 30 und 40 Nm und eine geringere Verlässlichkeit in sehr kleinen Drehmomentsbereichen. Diese sehr kleinen Drehmomentsbereiche wurden zuvor nicht untersucht, da Versuche an Probanden ausschließlich bei Abstufungen der Maximalkraft durchgeführt werden und sich somit in der Regel in höheren Bereichen befinden. Anhand dieser Daten ist eine Durchführung der QR Messmethode erst ab Drehmomenten von 30 Nm empfehlenswert und liefert dort Werte die im Fall der höchsten Genauigkeit um 20% abweichen. Zur Überprüfung der Testmethodik in höheren Drehmomentsbereichen ist eine Wiederholung der Testreihen mit Federn höherer Federkonstante sinnvoll.

Da es wie bereits in Abschnitt 5.2 erwähnt schwierig ist, den Einfluss von Methodik, Phantom und Auswertungsmethodik auf die Messwerte zu trennen, muss diese Erkenntnis auch mit den Daten von humanen Probanden wie aus den AGBRESA Studien verglichen werden. Hier kann der Einfluss des Phantoms subtrahiert und die Verlässlichkeit dieser Tendenz überprüft werden.

Weiterhin wurde die Intervall-Standardabweichung über dem angelegten Drehmoment aufgetragen (Abbildung 4.4). Die Intervall-Standardabweichung ist ein Maß für die Qualität der vorliegenden Daten, da sie untersucht wie sich die Steifigkeiten bei Verschieben des Untersuchungsintervalls verändern. Unter optimalen Bedingungen ist die Intervall-Standardabweichungen minimal und jedes Auswertungsintervall liefert die gleiche Steifigkeit. Für alle Federn und Konfigurationen steigen bei verringertem Drehmoment die Intervall-Standardabweichungen. Das heißt die in Abbildung (4.2) dargestellten Steifigkeiten wurden aus fünf Werten zusammengesetzt, die mehr um den Mittelwert streuen. Bei geringen Drehmomenten ist die Gefahr, dass Ausreißer die Werte beeinflusst haben, also höher. Messungen bei hohen Drehmomenten zeigen geringe Intervall-Standardabweichungen und die durch das Python Skript ermittelten fünf Steifigkeiten waren jeweils näher beieinander. Weit auseinander liegende Steifigkeiten innerhalb eines einzelnen QRs können nur durch eine schnell variierende Winkelposition oder -beschleunigung verursacht sein, die Messung weist also Schwingungen auf. Diese Schwingungen bei kleineren Drehmomenten wurde auch im Messprotokoll der Versuche dokumentiert. Dies könnte eine Begründung für weiter abweichende Steifigkeitswerte vom Erwartungswert bei kleinen Drehmo-

menten darstellen.

Abbildung 4.5 bietet eine Übersicht über die Variationskoeffizienten der Messungen ohne Dämpfer. Die Variationskoeffizienten wurden aus den Steifigkeitswerten der drei aufeinanderfolgenden QR Messungen ermittelt. Der Variationskoeffizient ist im Gegensatz zur Intervall-STD ein Maß der Reliabilität der Messmethode, wohingegen die Intervall-Standardabweichung lediglich über die Qualität der Daten Auskunft geben kann.

Die Variationskoeffizienten bewegen sich für alle Drehmomente zwischen 3 und 27%, ausgenommen bei kleinen angelegten Drehmomenten, wo V einen Wert von bis zu 170% annimmt. Bei kleinen Drehmomenten von <20 Nm zeigt eine Wiederholung also eher andere Messwerte als bei höheren Drehmomenten. Dies hängt in gewissem Maße aber auch mit den stärkeren Vibrationen und Schwingungen des Systems bei geringen Drehmomenten zusammen und kann nicht vollständig auf die Messmethodik zurückgeführt werden.

Schlussendlich wurden alle Messungen auch mit einem parallel geschalteten Dämpfer durchgeführt und ebenso Steifigkeiten, Abweichungen und Variationskoeffizienten über dem Drehmoment aufgetragen. Es wurde festgestellt, dass selbst der Steifigkeits-Messwert, der die kleinste Abweichung zur entsprechenden Drehfederkonstante zeigt, um 70% abweicht. Daraus ist zu schließen, dass die Dämpfungskonstante des Dämpfers zu hoch gewählt wurde und die Steifigkeiten der Federn nicht mehr differenziert werden können. Die Ergebnisse der Steifigkeitsmessungen mit Dämpfer erscheinen dadurch redundant.

Entsprechend wurde der Fokus dieser Arbeit bezüglich der Quick Release Testreihen auf die Auswertung und Diskussion der Ergebnisse ohne Einfluss eines Dämpfers gelegt.

Es ist jedoch festzuhalten, dass auch bei den Messungen mit Dämpfer die Variationskoeffizienten einen ähnlichen Trend wie die der Messungen ohne Dämpfer zeigen. Die einzelnen Wert sind zwar deutlich zu hoch, variieren innerhalb der jeweils drei aufeinander folgenden QRs bei hohen Drehmomenten zwischen 0 und 30%, also in einem ähnlichen Bereich wie bei Messungen ohne Dämpfer. Ab einem angelegten Drehmoment von 25 Nm treten drei Messwerte mit einem Variationskoeffizienten über diesem Wert auf, hier liegt V bei bis zu 90%.

5.5.3 Einfluss des voreingestellten Trägheitsmoment im MARES

Vor Diskussion des im MARES eingestellten Trägheitsmoments ist festzuhalten, dass keine der Messungen mit dem eigentlichen Trägheitsmoment der Kombination aus SLA und Phantom-Aufsätze durchgeführt wurde. Das im MARES für humane Probanden voreingestellte Trägheitsmoment des SLA beträgt $0,35\text{kgm}^2$ und wurde auch für die AGBRESA Studie genutzt. Dieses Trägheitsmoment ist auch das Moment, welches für die alle Messungen genutzt wurde, außer für jene Messungen, bei denen explizit der Einfluss des voreingestellten Trägheitsmoments evaluiert werden sollte. Die Information der Möglichkeit das Trägheitsmoment im MARES System zu verändern, ergab sich erst nach Durchführung eines Großteils der Messungen und zeitliche Einschränkungen

verhinderten eine Wiederholung aller Messungen. Daher wurde das Trägheitsmoment des neuen Aufbaus bestimmt und gezielt eine Messreihe mit einem voreingestellten Trägheitsmoment von $0,097\text{kgm}^2$ wiederholt. Bei Errechnung des neuen Trägheitsmoments von $0,097\text{kgm}^2$ sind Rechen- und Messfehler aufgetreten, sodass auch das Trägheitsmoment von $0,097\text{kgm}^2$ nicht dem eigentlichen Trägheitsmoment des Armes entspricht. Das wahre Trägheitsmoment des short levers plus das Trägheitsmoment des Phantoms wurde erst nach Fertigstellung aller Messreihen richtig bestimmt (vgl. Abschnitt 3.4).

Das Trägheitsmoment wird durch das genutzte Pythonskript über Formel 3.17 berechnet und hängt damit von der maximalen Beschleunigung und dem angelegten Drehmoment ab. Das Trägheitsmoment wird neben der Steifigkeit, dem Drehmoment und der Beschleunigung vom Python Skript für jeden einzelnen QR ausgegeben (vgl. Tabelle 3.4). Zur Überprüfung, ob die Einstellung des Trägheitsmomentes an der Drehachse in der MARES Software einen Einfluss auf die ausgegebene Trägheit, und damit über Formel 3.18 auch auf die Messung der Steifigkeit hat, wurde die Messreihe der Feder K_{248} ohne Dämpfung mit einem anderen voreingestellten Trägheitsmoment wiederholt.

Die erste Messreihe mit Feder K_{248} und einem eingestellten Moment von $0,35\text{ kgm}^2$ weist einen Mittelwert des Trägheitsmomentes von $(0,247 \pm 0,023)\text{ kgm}^2$ auf. Die zweite Messreihe mit dieser Feder mit einem auf $0,096\text{ kgm}^2$ geänderten Trägheitsmoment weist ein mittleres Moment von $0,296 \pm 0,008)\text{ kgm}^2$ auf. Die ermittelten Momente für ein niedrigeres voreingestelltes Moment liegen alle über dem höchsten Wert der Trägheit des hoch eingestellten Momentes. Zudem streuen die Daten des höheren voreingestellten Moments mehr um den Mittelwert. Bei kleiner eingestellten Momenten des SLA und Anhang an der Drehachse ergeben sich in den vorliegenden Daten höhere Trägheitsmomente, dies sollte nach Formel 3.18 bei gleichbleibenden anderen Parametern zu höheren Steifigkeiten führen. Die Steifigkeiten von $K_{248\text{neu}}$ liegen laut Tabelle 4.3 jedoch im Durchschnitt und für alle ausgewerteten Drehmomentbereich unter denen von K_{248} . Um diese Diskrepanz in Höhe des Trägheitsmoments und Höhe der Steifigkeit zu erklären, muss man die Zusammensetzung der entsprechenden Messwerte analysieren. Das Trägheitsmoment errechnet sich aus angelegtem Drehmoment und maximaler Beschleunigung nach Formel 3.17. Wenn das Trägheitsmoment also höher ist, ist auf einen höheren Drehmoment, eine niedrigere maximale Beschleunigung oder eine Kombination aus beidem, während der QR Testreihen mit K_{248n} im Vergleich zu K_{248} zu schließen.

Das MARES besitzt Regelmechanismen, die beispielsweise den SLA anhand des eingestellten Trägheitsmomentes in Gleichgewicht halten. Auch während der Versuche spielen diese Regelmechanismen eine Rolle. Es ist denkbar, dass bei einem kleiner eingestellten Moment das MARES weniger gegenregelt und somit höhere Drehmomente erzeugt werden. In einem Idealen System ist das Trägheitsmoment konstant und sollte nicht vom Drehmoment abhängen, da mit einem erhöhten Drehmoment auch die maximale Beschleunigung steigt. Eine Begründung könnte darin liegen, dass die Drehmomente durch eine geringere Gegenregelung bei einem niedrig eingestelltem Trägheitsmoment zwar den Drehmoment im Versuch erhöhen, das MARES eine Erhöhung

der maximalen Beschleunigung aber durch interne Mechanismen verhindert. Ein Beispiel für einen solchen Mechanismus ist das Gegensteuern des MARES in der Nähe der Endstops um diese nicht zu schädigen. Eine MARES interne Dämpfung der Beschleunigung zum Schutz der Probanden ist eine mögliche Erklärung.

Wie zuvor erwähnt steigt die Steifigkeit für K_{248neu} trotz erhöhtem ermitteltem Trägheitsmoment nicht. Daraus lässt sich schließen, dass der Term $\frac{\Delta\Theta''}{\Delta\Theta}$ der Formel 3.18 für K_{248n} kleiner ist. Also wird bei erhöhtem Trägheitsmoment entweder weniger beschleunigt, was mit der vorangegangenen Annahme der MARES Bremsung übereinstimmen würde, oder aber die Differenz der Winkelposition ist höher, was auf einen erhöhten Drehmoment hindeutet. Letzteres sollte in der Theorie aber wieder die maximale Beschleunigung erhöhen.

Zusammenfassend ist festzuhalten, dass die Einstellung des Trägheitsmoments anhand der vorliegenden Daten einen Einfluss auf die Messwerte der Steifigkeit nimmt. Beobachtungen des short lever-Verhaltens ergaben zusätzlich, dass ein Trägheitsmoment, welches näher am wahren Wert des Moments für den short lever + Fußteils liegt, zu geringeren Vibrationen und ungewünschten Schwingungen führt.

Die hier getroffenen Annahmen basieren nur auf den vorliegenden 12 Datenpunkten. Weiterhin wurde wie Eingangs erwähnt, keine Messreihe mit dem eigentlichen voreingestellten Trägheitsmoment des SLA und Phantom durchgeführt. Da ein Einfluss bei Vergleich der Steifigkeiten und ermittelten Trägheitsmomenten von K_{248} und K_{248neu} aber gegeben erscheint, erscheint eine weitergehende Evaluierung bezüglich dieser Einstellung sinnvoll.

5.5.4 Federsteifigkeiten der AGBRESA Studie

Es wurde exemplarisch für acht Probanden der AGBRESA Studie die Muskulotendinöse Steifigkeit des Plantarflexoren *M. triceps surae* mittels des Quick Release Test ermittelt. Von acht Probanden zeigten dabei zwei (G, H) eine negative Korrelation zwischen Drehmoment und Muskelsteifigkeit wie sie in der Literatur beschrieben wird (vgl. Abschnitt 5.5.1). Im Gegensatz zu den Steifigkeiten der Federn, die bei niedrigen Drehmomenten die höchsten Steifigkeiten zeigten, ist hier also ein Zusammenhang zwischen hohen Drehmomenten und hohen Steifigkeiten zu erkennen. Alle anderen Probanden zeigten keine oder eine niedrige Korrelation mit keiner ausreichenden Signifikanz ($p > 0.05$). Die Höhe der ermittelten Steifigkeiten stimmt jedoch mit den Daten aus der Literatur überein[13].

5.5.5 Federsteifigkeiten mit Auswertung durch Cosinus-Fit-Annäherung

In Abschnitt 4.2.4 wurden die Steifigkeiten der vier vermessenen Federn sowie der AGBRESA Probanden G und H über dem Drehmoment aufgetragen. Dabei wurden die Steifigkeiten auf zwei unterschiedliche Arten ausgewertet. Die erste Methode der Auswertung entspricht der herkömmlich bei QR genutzten Auswertung bei der der Stream der Winkelposition zweifach abgeleitet wird und aus Winkelposition, Winkelbeschleunigung und Drehmoment die Steifigkeit

direkt über Formel 3.17 und Formel 3.18 errechnet wird. Die zweite Auswertungsmethode, die im Verlaufe der Auswertung dieser Arbeit getestet wurde, legt eine Cosinus-Fit Funktion an die Daten des Positionstreams an, der Rest der Auswertungsmethodik erfolgt wie zuvor, nur das in diesem Falle die Cosinus Funktion im Auswertungsintervall abgeleitet wird. Die beiden Methoden werden im Folgenden evaluiert.

In fast allen Messreihen zeigt die Auswertungsmethode per Cosinus Fit geringere Messwerte als mit der herkömmlichen Methode. Die Steifigkeit für K_{80} liegt zwar höher, durch schlechtes Anliegen der Fit Funktion und der generell schon kleinen Datenlage für die Feder K_{80} handelt es sich jedoch nur um die Daten eines einzelnen QRs. Die andere Feder mit steigender Steifigkeit ist K_{248neu} . Da hier aber ein anderes Trägheitsmoment voreingestellt wurde, das sehr wahrscheinlich einen Einfluss auf die Steifigkeit ausübt (vgl. Abschnitt 5.5.3), ist der Vergleich zu dieser Federn unzulässig. Alle anderen Federn zeigen eine kleinere Steifigkeit durch Cosinus Fit. Die beiden ausgewerteten Probanden der AGBRESA Studie zeigen zudem eine geringere Standardabweichung der Steifigkeitsmesswerte als zuvor, die Steifigkeiten streuen demnach weniger bei der Cosinus-Fit-Methode.

Bei den AGBRESA Probanden ist nicht festzustellen, ob die reellen Werte eher zu hoch oder zu niedrig sind, daher ist die Höhe der Steifigkeit alleine kein Qualitätsmerkmal für die Auswertungsmethodik. Die geringere Standardabweichung jedoch spricht für den Cosinus Fit, da eine starke Streuung der Steifigkeiten im Hinblick auf vergangene Literatur nicht die Regel ist. Zur signifikanten Beurteilung müssten jedoch mehr Messdaten ausgewertet werden.

Bei Messung der Federsteifigkeiten liefert die Cosinus-Fit-Methode besonders im Hinblick auf die Steifigkeiten der geringen Drehmomente (10-30 Nm) Steifigkeiten, die Näher an der Drehfederkonstante der Feder und damit genauer sind. Das hängt jedoch elementar damit zusammen, dass Steifigkeiten für geringe Drehmomente eher zu hoch sind und die Fit Methode die generell geringere Steifigkeiten erzielt, damit grundsätzlich bessere Ergebnisse erzielt.

Zusammenfassend ist festzuhalten, dass es Unterschiede zwischen beiden Auswertungsmethoden, insbesondere im Hinblick auf die durchschnittliche Höhe der Steifigkeiten gibt. Folglich sollten beide Methoden weiter an mehr humanen Probanden und anderen Federn evaluiert werden, um festzustellen, welche Methodik verlässlicher signifikante Ergebnisse produziert. Vor Allem im Hinblick auf kommende Studien zur Muskelsteifigkeit ist es relevant, dass die Auswertungsmethodik weitergehend validiert wird.

5.6 Sinusoidal Perturbation

Bevor die Ergebnisse der SP-Untersuchungen diskutiert werden, werden zunächst generelle Probleme im Bezug auf die Testung erläutert. Zunächst ist hier zu nennen, dass die Untersuchungen mit der Feder K_{478} nur bedingt aussagekräftig sind. Da die Feder eine sehr hohe Federkonstante hat, ist nur eine sehr geringe Auslenkung der Feder nötig, um die gewünschten Drehmomente zu erreichen. Hierbei tritt bei geringen geforderten Drehmomenten das Problem auf, dass die

Feder so geringfügig gespannt wird, dass sie während der sinusoidalen Schwingungen der Tests regelmäßig völlig ungespannt ist. Auf Grund dieser Tatsache sind die Messdaten bei geringen Drehmomenten bei Verwendung der Feder K_{478} als unzuverlässig einzustufen. Sie wurden dennoch in der Auswertung der Ergebnisse berücksichtigt.

5.6.1 Phantominterne Analyse

Im Rahmen der Untersuchung des Einflusses der Eingangsvariablen kann nur eine relative Aussage getroffen werden. Da keine etablierte Referenz existiert, ist über die Standardabweichung lediglich die Aussage möglich, dass eine Eingangsvariable größeren oder geringeren Einfluss als eine andere Eingangsvariable hat. Auf dieser Grundlage lassen sich die Ergebnisse aus Kapitel 4.3.1 interpretieren. Hierbei zeigt sich, dass im Bezug auf den Gain des Systems, sowohl %MVC als auch MVC einen eher geringen Einfluss auf das Resultat haben ($\pm 0,4$ dB bzw. $\pm 0,3$ dB). Dies ist auch in den entsprechenden Diagrammen (Abb. 4.14 bzw. Abb. 4.13) ersichtlich, da die Mittelwerte der Gruppen hier sehr nah beieinander liegen. Einen, an der Standardabweichung gemessenen, drei- bis viermal größeren Einfluss haben die Drehfederkonstante K und die Dämpfungskonstante B ($\pm 1,4$ dB bzw. $\pm 1,0$ dB). Auch dies spiegelt sich in den Diagrammen wieder, da die entsprechenden Mittelwerte der Gruppen eindeutig voneinander abweichen (s. Abb. 4.15 bzw. Abb. 4.16). Die Aussage, die auf Grund dieser Untersuchungen im Bezug auf den Gain getroffen werden kann, ist, dass sowohl die MVC, welche zu Beginn der Testung mit dem MARES-System vorgenommen wird, sowie die prozentuale Abstufung %MVC nur einen relativ geringen Einfluss auf das resultierende Drehmoment unseres Phantom-Modells hat, dies entspricht den Erwartungen gemäß der mathematischen Modellierung. Bedeutender für das resultierende Drehmoment ist die Drehfederkonstante K und die Dämpfungskonstante B, auch dies deckt sich mit den Erwartungen.

Weiterhin wird nun die Phasenverschiebung betrachtet. Sowohl visuell als auch anhand der Standardabweichungen lässt sich feststellen, dass die Dämpfungskonstante B den größten Einfluss hat (Standardabweichung $\pm 21^\circ$). Im Gegensatz zu dem Einfluss auf den Gain, sind keine auffälligen Unterschiede zwischen K, MVC und %MVC festzustellen ($\pm 14^\circ$ bzw. $\pm 6,9^\circ$ bzw. $\pm 10^\circ$). Dass keine Unterschiede zwischen %MVC und MVC erkennbar sind, macht Sinn, da diese Unterscheidung im Falle unseres Phantoms eher künstlicher Natur ist. Die Angaben könnten durch eine Unterscheidung nach Drehmoment zusammengefasst werden, hierauf wird aus Gründen der Vergleichbarkeit mit Humanprobandendaten, in denen die prozentuale Muskelaktivierung durchaus eine Rolle spielt (siehe folgendes Kapitel), verzichtet.

5.6.2 AGBRESA-interne Analyse

Im Rahmen der Untersuchungen der AGBRESA-Daten konnte nur nach MVC und %MVC unterschieden werden, da keine Informationen zu B oder K vorliegen. Die Aussage des vorherigen Abschnittes, dass %MVC nur einen geringen Einfluss hat, konnte im Rahmen der AGBRESA-

Daten nicht bestätigt werden. Hier ergibt sich für den Gain eine Standardabweichung von $\pm 1,1$ dB. Überraschenderweise konnte jedoch kein erheblicher Unterschied innerhalb des Vergleiches der MVC-Werte festgestellt werden (Standardabweichung $\pm 0,3$ dB). Diese Ergebnisse der AGBRESA-Untersuchungen sind allerdings nur bedingt aussagekräftig, da für $MVC = 80$ Nm lediglich ein Datensatz zu Verfügung steht, für $MVC = 120$ Nm sind es sechs. Die AGBRESA-Daten wurden vor Beginn dieser Arbeit aufgenommen, weshalb uns zwar eine recht umfangreiche Datengrundlage zur Verfügung gestellt wurde, die MVC sind allerdings tatsächliche Maximalwerte und bewegten sich eher im Bereich zwischen 110 bis 190 Nm. Nur ein Datensatz lag im Bereich von 10 % um die im Rahmen der Phantomuntersuchungen vorgegebenen 80 Nm MVC. Bezüglich der Phasenverschiebung zeigt sich eine äußerst geringe Abweichung im %MVC- sowie im MVC-Vergleich ($\pm 5,6^\circ$ bzw. $\pm 3,0^\circ$)

5.6.3 Überprüfung der Validität des mathematischen Modells

Auf Grund dieser Erkenntnisse wurde nun die eigentliche Validierung der Messmethodik durchgeführt. Die Ergebnisse des Kapitels 4.3.2 bilden hierfür die Datengrundlage. In den Diagrammen dieses Kapitels sind alle Permutationen aus den drei Drehfederkonstanten K und den beiden Dämpfungskonstanten B dargestellt. Zusätzlich wird auch das mathematische Modell aufgetragen. Bei perfekter Übereinstimmung der praktisch erhobenen Messdaten und des theoretisch zu erwartenden Verhaltens, wären die entsprechenden Graphen perfekt deckungsgleich und die Validität des zugrundeliegenden mathematischen Modells zu 100 % bestätigt. Ein Vergleich absoluter Werte ist, wie schon bei den phantominternen Analysen in Kapitel 5.6.1, auch in diesem Fall nicht möglich, da keine verifizierte Referenz vorliegt.

Ein erster Schritt zur Validierung der Testmethodik ist die qualitative Bewertung der Ergebnisse. Betrachtet man die Ergebnisse aus Abbildung 4.19 für $B = 0$ Nmsrad⁻¹, so erkennt man eine eindeutige Diskrepanz der Kurvenverläufe der Phantom-Messdaten und des mathematischen Modells. Insbesondere fällt auf, dass bei Betrachtung des Gains, die mathematisch zu erwartende Resonanzkatastrophe bei Frequenzen zwischen 4 bis 8 Hz (je nach Drehfederkonstante) nicht in den experimentell ermittelten Phantom-Daten wiederzufinden ist. Die Kurven für Phantom und Modell fallen jedoch nach den erwarteten Resonanzpeaks in die gleiche Größenordnung und sind in ihren Tendenzen vergleichbar. Dies ist eine vielversprechende Erkenntnis, die auf eine grundsätzliche Vorhersagefähigkeit des Modells schließen lässt. Offen bleibt allerdings die Frage, wie weitreichend diese Vorhersagefähigkeit ist.

Bei Betrachtung der Phasenentwicklung von Phantom und Modell wird ebenfalls eine eindeutige Abweichung der beiden klar. Während laut Modell ein abrupter Umschlag der Phase von 0° auf -180° zu erwarten wäre, ist in den realen Daten ein langsamer Abfall der Phase erkennbar. Der abrupte Umschlag des mathematischen Modells korreliert mit dem erwarteten Resonanzpeak im Gain-Diagramm. Es könnte vermutet werden, dass in den Kurvenverläufen der Phase des Modells der Start des langsamen Abfalls mit der Resonanzfrequenz des Modell korreliert, allerdings erlaubt die zu geringe Auflösung der Datenpunkte keine zuverlässige Aussage dieser Art.

Eine Extrapolation der Phasen-Daten ist nicht zielführend, da der Verlauf der entsprechenden Kurven sich hierfür nicht eignet.

Sowohl im Bezug auf den Gain als auch auf die Phase liegt eine dämpfende Eigenschaft, die nicht vom Phantom selber stammt, sehr nahe, denn es sind zahlreiche Parallelen zum Einfluss einer Dämpfung im Rahmen einer Low-Pass-Übertragungsfunktion zweiter Ordnung (wie wir sie verwenden) auf den Kurvenverlauf erkennbar. Da die Experimentaldaten der betrachteten Diagramme ohne Dämpfer erfolgen ($B = 0 \text{ Nmsrad}^{-1}$, sollte keine Dämpfung erkennbar sein, da sie allerdings vorliegt, muss das MARES eine dämpfende Wirkung auf das System haben. Der Einfluss einer Dämpfung auf Gain und Phase eines Systems zweiter Ordnung ist in folgender Abbildung 5.1 dargestellt. Hierbei gilt: $Q = \frac{1}{2\zeta}$, wobei ζ das Dämpfungsverhältnis ist. Dies lässt sich auf die während dieser Arbeit verwendete Übertragungsfunktion 3.21 übertragen.

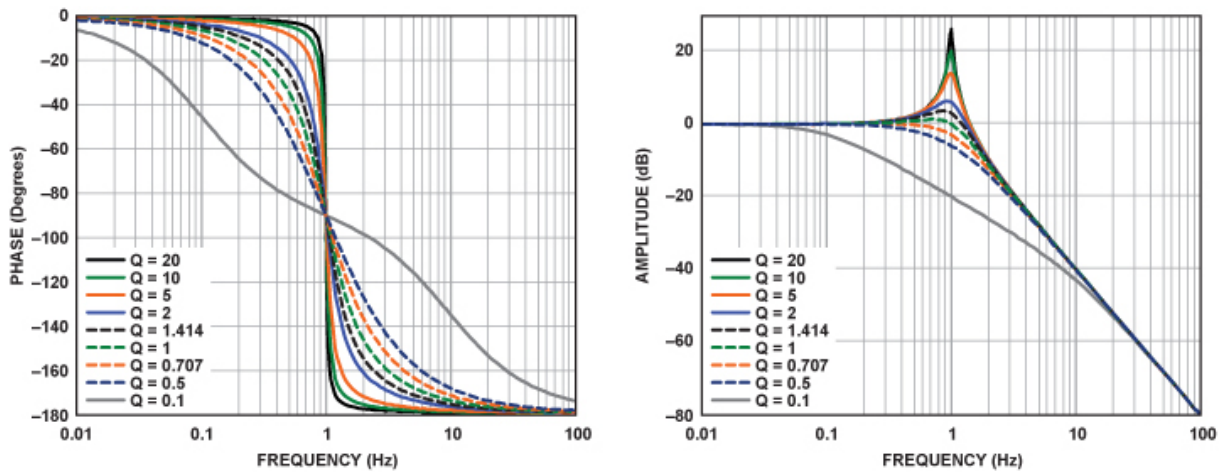


Abbildung 5.1: Beispielhafte Darstellung des Einflusses von Dämpfung auf den Gain und die Phase eines Systems zweiter Ordnung. Je geringer Q ist, desto höher ist die Dämpfung. Amplitude und Gain sind Synonyme. Quelle: [80]

Aus diesen Beobachtungen folgt die Frage, warum die Abweichungen, insbesondere im Bezug auf den fehlenden Resonanzpeak, vorliegen und welche Komponente fehlerhaft ist. Hierbei ist die mathematische Modellierung die Referenz, welche vorausgesetzt wird und anhand derer gemessen wird. Es bleibt also das Phantom als Fehlerquelle, das Interface zwischen Phantom und MARES oder das MARES selbst. Unser Phantom wurde bewusst physikalisch und technisch so simpel gehalten, wie es möglich war, während es gleichzeitig noch die Anforderungen des mathematischen Modells erfüllt. Aufgrund des Designs des Phantoms ist nicht davon auszugehen, dass hier derartige Probleme auftreten, die die resultierende Diskrepanz zwischen mathematischem Modell und experimentell ermittelten Daten erklären könnten.

Die erste mögliche Fehlerquelle ist dann die Schnittstelle zwischen Phantom und MARES, also die Befestigung. Diese weist jedoch keine so erheblichen Schwächen auf, dass sie die Abweichungen verursachen könnten, vor allem ist keine dämpfende oder glättende Wirkung durch die Befestigung zu erwarten. Dennoch gibt es hier noch Raum für weitere Verbesserungen, um die-

se mögliche Fehlerquelle endgültig auszuschließen. Die lückenlose und passgenaue Verbindung zwischen Phantom und MARES sollte in zukünftigen Testaufbauten besonders berücksichtigt werden, um ein negativen Einfluss auf die Qualität der Ergebnisse zu eliminieren.

Schließlich bleibt das MARES selbst als Ursache für die fehlende Übereinstimmung von Experimentdaten und mathematischem Modell. Hinsichtlich der zahlreichen Probleme und Irregularitäten, die während der praktischen Arbeit mit dem MARES auftraten, ist davon auszugehen, dass hier am ehesten der Grund für die Diskrepanzen zu finden ist. Die Mechanismen zum Schutz von Mensch und Maschine könnten hierbei besonders relevant sein, da so z.B. ungedämpften Schwingungen aktiv entgegengewirkt wird. Die Problematik ist hierbei, dass das MARES bis zu einem bestimmten Punkt eine Blackbox ist, deren Verhalten im Rahmen dieser Arbeit nicht erarbeitet werden konnte, da dies mit unverhältnismäßig hohem zeitlichen Mehraufwand verbunden gewesen wäre. Hinweise auf ein Eingreifen im Sinne einer MARES-internen Dämpfung findet sich in der Abbildung 4.19. Hier weist die Kurve des mathematischen Modells einen eindeutigen Peak an der Resonanzfrequenz, also einen Resonanzkatastrophenfall, auf. Ein solcher Peak findet sich in keinsten Weise in den Phantomdaten wieder, was darauf schließen lässt, dass das MARES der zu erwartenden Resonanzkatastrophe durch eine interne Dämpfung entgegenwirkt, was sich in einer Glättung der Kurve niederschlägt. Hier macht ein Eingreifen seitens des MARES auch Sinn, um eine Gefährdung von Mensch und Maschine zu verhindern. Schon anhand dieser Tatsache wird deutlich, dass das MARES vermutlich intern eingreift und somit eine tiefgreifende Untersuchung des MARES-Verhaltens nötig ist. Diese These wird durch die langsame Phasenverschiebung gestützt, die ebenfalls auf eine nicht unerhebliche interne Dämpfung durch das MARES hindeutet.

Die Aussage, welche auf Grundlage der Daten der Extrapolation zu Frequenz 0 Hz (Abbildung 4.22 und Tabelle 4.7) getroffen werden kann, ist, dass das mathematische Modell das Phantomverhalten bei einer Frequenz von 0 Hz gut prognostizieren kann. Die Übereinstimmung der Steifigkeiten von Modell und Phantom bedeutet, dass das Auseinanderdriften von Modell- und Phantomkurve in den frequenzabhängigen Termen I und/oder B begründet ist. Dies unterstützt weiterhin die These, dass das MARES eine interne Dämpfung besitzt bzw. induziert, welche Einfluss auf die Messungen hat. Dies bedeutet im Umkehrschluss, dass bei Tests an Humanprobanden mit der Formel 2.1 nicht nur muskuläre Eigenschaften berücksichtigt werden, sondern auch die Eigenschaften des MARES mit einfließen. Somit ist es wahrscheinlich, dass mit der SP-Methodik am MARES aus gemessenen Drehmomenten nicht nur auf muskuloartikuläre Eigenschaften zurückgeschlossen werden.

Die Erkenntnisse der SP-Untersuchungen mit Dämpfer $B = 105 \text{ Nmsrad}^{-1}$, wie sie in Abbildung 4.20 zu sehen sind, zeigen eine ähnliche Situation wie die Situation für $B = 0 \text{ Nmsrad}^{-1}$, allerdings wird die Interpretation der Daten durch eine zusätzliche Dämpfung in Form unseres Dämpfers zusätzlich verkompliziert. Was zu erkennen ist, ist, dass die Phantomdaten in

noch größerem Maße von dem mathematisch zu erwartenden Modell abweichen. Dies wird zum Teil auf die ungünstig hohe Dämpfungskonstante zurückzuführen sein. Im Gegensatz zur Situation ohne Dämpfer unterscheiden sich sowohl die Werte für Gain, als auch für Phase stark von der mathematischen Modellierung. Auch die generellen Formen der Kurven unterscheiden sich erheblich, die Gründe hierfür gestalten sich noch ungewisser als für die Messungen ohne Dämpfer. Deswegen ist für eine sinnvolle Bewertung der SP-Tests mit Hilfe unseres Phantoms mit Dämpferzunächst ein vollständiges Verständnis des MARES-Verhaltens nötig.

5.7 Vergleich der Steifigkeiten zwischen Modell, QR und SP

Um eine Empfehlung bezüglich der zu präferierenden Messmethode zur Bestimmung von Steifigkeiten mit Hilfe des MARES aussprechen zu können, werden im Folgenden die Ergebnisse der Steifigkeitsbestimmung aus QR und SP mit dem mathematischen Modell verglichen. In der nachstehenden Tabelle sind alle relevanten Ergebnisse aufgelistet.

Tabelle 5.1: Vergleich der federabhängig ermittelten Steifigkeitswerte aus SP bei $B = 0 \text{ Nms/rad}$ (S_{SP0}) bzw. QR (S_{QR10} bis S_{QR60}) mit den mathematisch zu erwartenden Ergebnissen (S_{Modell}). Die Ergebnisse der QR-Tests (Spalten) sind nach Drehmomenten aufgeschlüsselt (angezeigt durch die Zahl im Index, z.B. S_{QR10} = Steifigkeitswert, QR-Testung, 10 Nmrad^{-1}). In den beiden letzten Spalten der Tabelle ist die Differenz von S_{SP} zu S_{Modell} , bzw die Differenz des besten QR-Wertes zu S_{Modell} dargestellt. Der beste QR-Wert bezeichnet den Wert aus S_{QR10} bis S_{QR60} , der die geringste Differenz zu S_{Modell} aufweist. Alle K-Werte sind in Nmrad^{-1} angegeben. Zeilenweise werden die Ergebnisse entsprechend der Input-Federkonstanten aufgelistet. Hierbei weisen die Indizes die entsprechende Drehfederkonstante der eingesetzten Feder aus. K_{248n} bezeichnet die QR-Messungen mit der K_{248} -Feder mit veränderter Trägheitsmoment-Einstellung am MARES. Die entsprechenden Werte der SP-Messungen fehlen, da kein Unterschied zwischen den Trägheitsmomenteneinstellungen während der Tests beobachtet werden konnte.

	S_{Modell}	S_{SP0}	S_{QR10}	S_{QR20}	S_{QR30}	S_{QR40}	S_{QR60}	$\Delta S_{SP0} S_{Modell} [\%]$	$\Delta S_{QR} S_{Modell} [\%]$
K_{80}	80		47	46	57				-29
K_{158}	158	150	386	102	113	118	134	-5	-15
K_{248}	248	251		444	229	207	173	1	-8
K_{248n}	248			285		160	141		21
K_{478}	478	562		815	527	442	363	18	-8

In Tabelle 5.1 finden sich die prozentualen Abweichungen der Steifigkeitswerte der SP-Messungen und der QR-Messungen von dem mathematischen Modell. Hierbei fehlen die SP-Werte für die leichteste Feder (K_{80}), da diese nicht im Rahmen der SP-Messungen verwendet werden konnte. Außerdem fehlen die SP-Werte für K_{248n} , da die veränderte Trägheitsmomenteinstellung am MARES keine Auswirkungen auf die Ergebnisse der SP-Messung zeigte. Somit bleiben zur Untersuchung der Eignung der Testmethodik die Δ -Werte für K_{158} , K_{248} und K_{478} . Hierbei zeigt sich, dass die SP-Tests sowohl im Falle von K_{158} , als auch bei K_{248} , Ergebnisse für die Steifigkeit hervorbringen, die näher an den mathematisch zu erwartenden liegen (-5 % vs. -15 % bzw. 1 % vs.

-8 %). Bei Untersuchungen mit der Feder K_{478} , lassen sich über die QR-Testmethode Ergebnisse erzielen, die näher an dem mathematischen Modell sind als bei der SP-Testung. Allerdings ist dies nur in den Bereichen eines angelegten Drehmoments von 30 bis 40 Nm der Fall. Außerdem sind die SP-Werte für K_{478} nur bedingt verlässlich aus Gründen, die bereits am Anfang des Kapitels erläutert wurde. Die QR-Messungen hingegen zeigen wie in Kapitel 5.5 erläutert, sehr unterschiedlich gut ausgeprägte Steifigkeiten für die unterschiedlichen angelegten Drehmomente. Durch die dadurch bedingte nötige Aufteilung der Messwerte nach Drehmomenten wird K_{478} jedoch nur durch wenige Werte repräsentiert. Das macht es unmöglich eine signifikante Aussage zur Verlässlichkeit dieser Daten, die über eine Tendenz hinausgeht, zu treffen.

Für Federsteifigkeiten und damit Muskelsteifigkeiten von 158 und 248 Nmrad^{-1} ist unabhängig von angelegten Drehmomenten auf Basis dieser Daten die Messung durch Sinusoidal Perturbation die bessere Messmethode und sollte zukünftig bei Steifigkeitsmessungen präferiert werden. Im Bereich geringer Steifigkeiten kann durch fehlende SP Daten keine Aussage getroffen werden. Im Bereich hoher Steifigkeiten liefert die QR-Messmethode bessere Ergebnisse, wenngleich die Datenlage hier nicht ausreicht um eine Aussage mit ausreichender Signifikanz zu treffen.

Eine mögliche Erklärung für genauere Ergebnisse bei Messung mit Hilfe der SP-Messmethode im Vergleich zur QR-Messmethode ist die Asymmetrie des Quick Release. Die Bewegung erfolgt nur in Richtung der Plantarflexion und entsprechend werden die viskoelastischen Eigenschaften auch nur in diese Bewegungsrichtung abgebildet. Die Sinusoidal Perturbation Methode führt zur Flexion und Extension des *M. triceps surae* und bildet die Gesamtsteifigkeit des Muskels damit möglicherweise besser ab. Zudem erfolgen bei SP jeweils eine Vielzahl an Bewegungen, wohingegen QR jeweils nur eine Bewegung abbildet. Die QR-Messungen werden zwar mehrmals hintereinander wiederholt, eine durchgehende Messung mit mehreren Bewegungen wie bei SP ist jedoch potenziell weniger fehleranfällig.

5.8 Entwurf eines alternativen Messsystems

Da es nun zahlreiche Hinweise darauf gibt, dass die Untersuchung von Gelenkdynamiken an Humanprobanden am MARES nach unserer Methodik keine zufriedenstellenden Ergebnisse hervorbringt, gilt es alternative Möglichkeiten zu untersuchen, die einen robusteren Versuchsaufbau bieten. Hierbei ist vor allem ein fundamentaler Aspekt unserer Tests zu berücksichtigen. Und zwar, dass ein System genutzt wird, welches auf Grund seines primären Einsatzgebietes auf der ISS eine Fülle an Untersuchungs- und Trainingsmöglichkeiten bietet und das bei extrem platzsparender Bauweise. Während es so für den Einsatz auf der ISS ein ingenieurtechnisches Meisterwerk ist, ist es im Falle von Untersuchungen auf der Erde gewissermaßen überqualifiziert. Die Überlegung ist also, ein spezielles, alternatives System zu entwerfen, welches den Anforderungen und Wünschen im Rahmen der Vorgaben der Untersuchungen von Gelenkdynamiken am DLR gerecht wird, hierbei sollen die prinzipiellen Messmethodiken QR und SP erhalten bleiben. Aus diesem Konzept ergeben sich zwei signifikante Vorteil/Nachteil-Paare.

Zum Einen ist das System auf eine Art der Testung spezialisiert. Es verringert sich also die Vielseitigkeit des Systems, gleichzeitig steigt aber die Genauigkeit sowie Zuverlässigkeit für die spezielle Art der Testung. Zum Anderen sinkt die Vergleichbarkeit mit den Ergebnissen, welche über das MARES erzeugt wurden und die äußerst wichtigen Daten von echten Astronauten umfasst, die auf Grund ihrer Seltenheit enorm wertvoll sind. Auch hier ist der Vorteil dieses Tradeoffs, dass eine sehr viel zielorientiertere Testung durchgeführt werden kann. Nachdem die zentralen Überlegungen nun dargelegt sind, wird im Folgenden erläutert, wie ein solches alternatives System aussehen könnte, welches eine möglichst schnörkellose und robuste Möglichkeit zur Untersuchung von Gelenkdynamiken bieten könnte.

Ein zentraler Bestandteil zur QR- und SP-Untersuchung ist der Aktuator, also der Motor des Systems. Dieser ist zwingend für die Testung nötig. Während das MARES einen Motor verwendet, der bis zu 900 Nm Drehmoment generieren kann, käme ein spezialisiertes Messsystem mit einem deutlich schwächeren Motor aus, der ca. 200 Nm aufbringen müsste. In zahlreichen Studien am MARES zeigte sich, dass Humanprobanden bei der Messung ihrer MVC selten über 200 Nm Drehmoment bei einer Plantarflexion erzeugen. Weiterhin kann der MARES-Motor Winkelgeschwindigkeiten von bis zu 515 °/s leisten, dieser Wert sollte in einem alternativen System maximiert werden, um einen Einfluss durch verzögerte Positionierung der Fußaufnahme zu minimieren. Hier liegen die Grenzen vor allem in den Kosten begründet, da, anders als beim MARES, räumliche Ausmaße kaum eine Rolle spielen. Weiterhin wird ein möglichst solides System zur Fixierung des Nutzers benötigt, um äußere Einflüsse wie Verschieben oder Vibrationen zu minimieren. Dies wäre über eine Verschraubung des Systems mit dem Boden nötig, welche beim MARES fehlt. Das Konzept zur Fixierung sollte eine unkomplizierte Verstellung der Gelenkwinkel des Benutzers, sowie eine Anpassbarkeit an die Physis bei gleichzeitiger größtmöglicher Bequemlichkeit möglich machen. Hier könnten Konflikte auftreten, welche durch einen entsprechenden Kompromiss ausgeräumt werden müssten. Wie auch der Sitz und seine Peripherie,

muss auch das Interface des Motors an den Probanden anpassbar sein. Ein weiterer zentraler Bestandteil eines solchen Messsystems wäre die Sensorik, welche möglichst feinfühlig alle relevanten Parameter erfassen sollte, hier bietet das MARES mit einer Abtastfrequenz von 2000 Hz einen sehr guten Richtwert, doch auch hier ist der Kostenfaktor die entscheidende, limitierende Größe. Der wichtigste Aspekt eines alternativen Systems wäre jedoch die Garantie der Sicherheit von Mensch und Maschine, somit müssten an allen relevanten Stellen neben softwareseitigen Failsafes auch hardwarebasierte installiert sein. Ein immenser Vorteil eines eigenen Messsystems liegt darin, dass die Software aus dem eigenen Haus käme und somit eine vollständige Dokumentation vorläge und damit einhergehend die Manipulation des Systemverhaltens möglich wäre, was im Falle der Arbeit mit dem MARES nur in sehr begrenzter Weise zutrifft.

Kapitel 6

Fazit und Ausblick

Zunächst ist festzuhalten, dass der erste Teil der ursprünglichen Frage der Thesis grundsätzlich erfolgreich beantwortet werden konnte. Es wurde erfolgreich ein Phantom zur Überprüfung der Validität der QR- und SP-Messmethoden am MARES konstruiert und die entsprechenden Untersuchungen bezüglich der Validität durchgeführt. Bezüglich des zweiten Teils der Frage, der Frage nach der Validität der QR- und SP-Untersuchungen konnte gezeigt werden, dass die praktischen Ergebnisse der QR- und der SP-Messmethode erheblich voneinander abweichen.

Die Messungen durch Quick Release generierten Ergebnisse der Steifigkeit, die in ihrer Tendenz deutlich von der vorhandenen Literatur abwichen. So zeigten sich bei geringen angelegten Drehmomenten hohe Steifigkeiten, was den Schluss nahelegt, dass die QR Messmethodik in diesen Bereichen nicht zulässig ist. Dafür sprechen ebenfalls die höheren Variationskoeffizienten in kleinen Drehmomentsbereichen sowie die hohen Intervall-Standardabweichungen als Maß für die Unregelmäßigkeit der Winkelposition während der Testdurchläufe. In Drehmomentsbereichen ab 30 Nm zeigten sich bessere Ergebnisse, obgleich auch hier die Abweichungen vom Erwartungswert bis zu 30% betrugen. Gleichzeitig ist jedoch festzuhalten, dass gängige Quick-Release-Messungen generell eher bei höheren Drehmomenten durchgeführt werden. Eine Wiederholung der durchgeführten Testreihen mit Drehmomenten nahe denen von humanen Probanden ist sinnvoll. Zudem wurde der Einfluss des voreingestellten Trägheitsmoments auf die Steifigkeit des Quick Release experimentell gezeigt. Dies ist eine wichtige Erkenntnis die zu besseren Ergebnissen in zukünftigen Studien führen kann. Vor allem die partielle Auswertung der AGBRESA-Daten hat gezeigt, dass die Auswertungsmethodik durchaus eine Rolle bei Versuchen der Steifigkeitsbestimmung spielt und in zukünftigen Studien die Auswertungsmethode validiert werden sollte.

Weiterhin konnte gezeigt werden, dass im Falle der SP-Messungen keine Übereinstimmung von Experimentaldaten und mathematisch zu erwartenden Daten vorliegt. Somit ist die Validität der Messmethoden zu diesem Zeitpunkt nicht zu bestätigen. Im Rahmen der Untersuchungen konnten jedoch zahlreiche Probleme identifiziert werden, welche eine zweifelsfreie Beantwortung der Frage nach der Validität der Methoden nicht zulassen.

Allerdings konnten aus den entsprechenden Untersuchungen auch eine Vielzahl an Erkenntnissen bezüglich der Funktionsweise des MARES, des Phantoms und der Testmethodiken gewonnen werden. Diese bilden eine hervorragende Grundlage um eine zukünftige, detailliertere Untersuchung der zentralen Hypothese dieser Arbeit zu ermöglichen, wodurch die zweifelsfreie

Beantwortung ermöglicht werden könnte. Hierfür sollten die beteiligten Systeme individuell genauer charakterisiert werden, um z.B. das Verhalten des MARES in Gänze nachvollziehen zu können. Auch im Bezug auf das Phantom konnten Verbesserungspotentiale definiert werden, welche weitere Unsicherheiten in der Durchführung eliminieren können. In Zukunft sollte also die Grundlage in Form des vollständigen Verständnisses des MARES gelegt werden, von da aus müsste zur Sicherheit eine weitere Validierung des Phantoms stattfinden, derzeit sind wir allerdings zuversichtlich, dass das Phantom in zukünftigen Studien eine sehr hilfreiche, technisch einwandfrei bestimmbare Referenz darstellen kann.

Danksagung

An dieser Stelle möchten wir denjenigen danken, die uns während des Studiums sowie während der Durchführung des Masterprojektes und der anschließenden Thesis begleitet und unterstützt haben.

Unser expliziter Dank geht an die Mitarbeiter der Abteilung Muskel- und Knochenstoffwechsel des Instituts Luft- und Raumfahrtmedizin des Zentrums für Deutsche Luft- und Raumfahrt für die herzliche Aufnahme in die Arbeitsgruppe und eine harmonische Arbeitsatmosphäre.

Im Speziellen danken wir Prof. Dr. Jörn Rittweger und Dr. rer. nat. Uwe Mittag für die umfangreiche Betreuung und fachliche Unterstützung der Masterarbeit. Ebenfalls danken wir Herr Wolfram Sies für die tatkräftige und fundierte Unterstützung in Fragen der Ingenieurskunst sowie den Mitarbeitern der DLR-Werkstatt für Ratschläge im Konstruktionsprozess des Phantoms und das Bereitstellen von Werkzeug und Materialien.

Des Weiteren gilt ein besonderer Dank Prof. Dr. Klaus Peikenkamp von der Fachhochschule Münster für die Betreuung und Unterstützung während der Masterarbeit.

Zuletzt danken wir unseren Familien und Freunden die uns während unseres Studiums unterstützt und dieses Studium möglich gemacht haben.

Eidesstattliche Erklärung

Wir versichern, dass wir die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und uns anderer als der in den beigefügten Verzeichnissen angegebenen Hilfsmittel nicht bedient haben. Alle Textstellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Alle Quellen, die dem World Wide Web entnommen oder in einer digitalen Form verwendet wurden, sind der Arbeit beigefügt. Der Durchführung einer elektronischen Plagiatsprüfung stimmen wir hiermit zu. Die eingereichte elektronische Fassung der Arbeit entspricht der eingereichten schriftlichen Fassung exakt. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht. Wir sind uns bewusst, dass eine unwahre Erklärung rechtliche Folgen haben kann.

Ort, Datum, Unterschrift

Ort, Datum, Unterschrift

Literatur

- [1] J. B. WEST. „Historical Aspects of the Early Soviet/Russian Manned Space Program“. In: *Journal of applied physiology (Bethesda, Md. : 1985)* 91.4 ((Jan. 2001)). Historical Article Journal Article Portrait, S. 1501–11. ISSN: 8750-7587. DOI: 10.1152/jappl.2001.91.4.1501.
- [2] P. D. HODKINSON u. a. „An Overview of Space Medicine“. In: *British journal of anaesthesia* 119.suppl_1 ((Jan. 2017)). Journal Article Review, S. 143–153. DOI: 10.1093/bja/aex336.
- [3] AJITKUMAR P. MULAVARA u. a. „Physiological and Functional Alterations after Spaceflight and Bed Rest“. In: *Medicine and science in sports and exercise* 50.9 ((Jan. 2018)). Journal Article Research Support, N.I.H., Extramural Research Support, U.S. Gov't, Non-P.H.S., S. 1961–1980. DOI: 10.1249/MSS.0000000000001615.
- [4] E. A. RADUGINA u. a. „Exposure to Microgravity for 30 Days Onboard Bion M1 Caused Muscle Atrophy and Impaired Regeneration in Murine Femoral Quadriceps“. In: *Life sciences in space research* 16 ((Jan. 2018)). Journal Article, S. 18–25. DOI: 10.1016/j.lssr.2017.08.005.
- [5] SCOTT TRAPPE u. a. „Exercise in Space: Human Skeletal Muscle after 6 Months Aboard the International Space Station“. In: *Journal of applied physiology (Bethesda, Md. : 1985)* 106.4 ((Jan. 2009)). Journal Article Research Support, U.S. Gov't, Non-P.H.S., S. 1159–68. ISSN: 8750-7587. DOI: 10.1152/japplphysiol.91578.2008.
- [6] AJITKUMAR P. MULAVARA u. a. „Locomotor Function after Long-Duration Space Flight: Effects and Motor Learning during Recovery“. In: *Experimental brain research* 202.3 ((Jan. 2010)). Journal Article Research Support, U.S. Gov't, Non-P.H.S., S. 649–59. DOI: 10.1007/s00221-010-2171-0.
- [7] SCOTT J. WOOD, WILLIAM H. PALOSKI und JONATHAN B. CLARK. „Assessing Sensorimotor Function Following ISS with Computerized Dynamic Posturography“. In: *Aerospace medicine and human performance* 86.12 Suppl ((Jan. 2015)). Journal Article Research Support, U.S. Gov't, Non-P.H.S., S. 45–53. ISSN: 2375-6314. DOI: 10.3357/AMHP.EC07.2015.

- [8] KARINA MARSHALL-GOEBEL, RAHUL DAMANI und ERIC M. BERSHAD. „Brain Physiological Response and Adaptation During Spaceflight“. In: *Neurosurgery* 85.5 ((Jan. 2019)). Journal Article Research Support, U.S. Gov't, Non-P.H.S. Review, S. 815–821. DOI: 10.1093/neuros/nyz203.
- [9] STUART M. C. LEE u. a. „Orthostatic Intolerance After ISS and Space Shuttle Missions“. In: *Aerospace medicine and human performance* 86.12 Suppl ((Jan. 2015)). Journal Article Research Support, U.S. Gov't, Non-P.H.S., S. 54–67. ISSN: 2375-6314. DOI: 10.3357/AMHP.EC08.2015.
- [10] ALAN D. MOORE u. a. „Peak Exercise Oxygen Uptake during and Following Long-Duration Spaceflight“. In: *Journal of applied physiology (Bethesda, Md. : 1985)* 117.3 ((Jan. 2014)). Journal Article Research Support, U.S. Gov't, Non-P.H.S., S. 231–8. ISSN: 8750-7587. DOI: 10.1152/japplphysiol.01251.2013.
- [11] M. W. BUNGO u. a. „Echocardiographic Evaluation of Space Shuttle Crewmembers“. In: *Journal of applied physiology (Bethesda, Md. : 1985)* 62.1 ((Jan. 1987)). Journal Article, S. 278–83. ISSN: 8750-7587. DOI: 10.1152/jappl.1987.62.1.278.
- [12] P. NORSK und M. EPSTEIN. „Manned Space Flight and the Kidney“. In: *American journal of nephrology* 11.2 ((Jan. 1991)). Editorial Research Support, Non-U.S. Gov't, S. 81–97. ISSN: 0250-8095. DOI: 10.1159/000168282.
- [13] DANIEL LAMBERTZ u. a. „Effects of Long-Term Spaceflight on Mechanical Properties of Muscles in Humans“. In: *Journal of Applied Physiology* 90.1 ((Jan. 2001)). Journal Article Research Support, Non-U.S. Gov't, S. 179–188. ISSN: 8750-7587. DOI: 10.1152/jappl.2001.90.1.179.
- [14] A. PAVY-LE TRAON u. a. „From Space to Earth: Advances in Human Physiology from 20 Years of Bed Rest Studies (1986-2006)“. In: *European journal of applied physiology* 101.2 ((Jan. 2007)). Journal Article Review, S. 143–94. ISSN: 1439-6319. DOI: 10.1007/s00421-007-0474-z.
- [15] LEONARD NĂSTASE, DRAGOS CRETOIU und SILVIA MARIA STOICESCU. „Skeletal Muscle Damage in Intrauterine Growth Restriction“. In: *Advances in experimental medicine and biology* 1088 ((Jan. 2018)). Journal Article Review, S. 93–106. ISSN: 0065-2598. DOI: 10.1007/978-981-13-1435-3-5..
- [16] GEORGE H. CARDINET. „Skeletal Muscle Function“. In: (), S. 407–440. DOI: 10.1016/B978-012396305-5/50017-8.
- [17] JAMES A. LOEHR u. a. „Physical Training for Long-Duration Spaceflight“. In: *Aerospace medicine and human performance* 86.12 Suppl ((Jan. 2015)). Journal Article Research Support, U.S. Gov't, Non-P.H.S., S. 14–23. ISSN: 2375-6314. DOI: 10.3357/AMHP.EC03.2015.

- [18] XIAONAN H. WANG. „MicroRNA in Myogenesis and Muscle Atrophy“. In: *Current opinion in clinical nutrition and metabolic care* 16.3 ((Jan. 2013)). Journal Article Research Support, N.I.H., Extramural Research Support, Non-U.S. Gov't Review, S. 258–66. DOI: 10.1097/mco.0b013e32835f81b9.
- [19] DANIEL LAMBERTZ u. a. „Influence of Long-Term Spaceflight on Neuromechanical Properties of Muscles in Humans“. In: *Journal of applied physiology (Bethesda, Md. : 1985)* 94.2 ((Jan. 2003)). Journal Article Research Support, Non-U.S. Gov't Journal Article Research Support, Non-U.S. Gov't, S. 490–8. ISSN: 8750-7587. DOI: 10.1152/japplphysiol.00666.2002.
- [20] INESSA B. KOZLOVSKAYA und ANATOLY I. GRIGORIEV. „Russian System of Countermeasures on Board of the International Space Station (ISS): The First Results“. In: *Acta astronautica* 55.3-9 ((Jan. 2004)). Journal Article, S. 233–7. ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2004.05.049.
- [21] A. LEBLANC u. a. „Muscle Volume, MRI Relaxation Times (T2), and Body Composition after Spaceflight“. In: *Journal of applied physiology (Bethesda, Md. : 1985)* 89.6 ((Jan. 2000)). Journal Article Research Support, U.S. Gov't, Non-P.H.S., S. 2158–64. ISSN: 8750-7587. DOI: 10.1152/jappl.2000.89.6.2158.
- [22] J. ZANGE u. a. „Changes in Calf Muscle Performance, Energy Metabolism, and Muscle Volume Caused by Long-Term Stay on Space Station MIR“. In: *International journal of sports medicine* 18 Suppl 4 ((Jan. 1997)). Journal Article, S. 308–9. DOI: 10.1055/s-2007-972738.
- [23] G. A. CAVAGNA. „Storage and Utilization of Elastic Energy in Skeletal Muscle“. In: *Exercise and sport sciences reviews* 5 ((Jan. 1977)). Journal Article Review, S. 89–129.
- [24] JAN BABIC und JADRAN LENARCIC. „In Vivo Determination of Triceps Surae Muscle-Tendon Complex Viscoelastic Properties“. In: *European journal of applied physiology* 92.4-5 ((Jan. 2004)). Clinical Trial Journal Article, S. 477–84. ISSN: 1439-6319. DOI: 10.1007/s00421-004-1107-4.
- [25] S. FUKASHIRO u. a. „Ultrasonography Gives Directly but Noninvasively Elastic Characteristic of Human Tendon in Vivo“. In: *European journal of applied physiology and occupational physiology* 71.6 ((Jan. 1995)). Journal Article, S. 555–7. ISSN: 0301-5548. DOI: 10.1007/bf00238560.
- [26] MICHEL POUSSON, JACQUES VAN HOECKE und FRANCIS GOUBEL. „Changes in Elastic Characteristics of Human Muscle Induced by Eccentric Exercise“. In: *Journal of biomechanics* 23.4 ((Jan. 1990)). PII: 0021929090900628, S. 343–348. ISSN: 0021-9290. DOI: 10.1016/0021-9290(90)90062-8.

- [27] I. W. HUNTER und R. E. KEARNEY. „Invariance of Ankle Dynamic Stiffness during Fatiguing Muscle Contractions“. In: *Journal of biomechanics* 16.12 ((Jan. 1983)). PII: 0021929083900994, S. 985–991. ISSN: 0021-9290. DOI: 10.1016/0021-9290(83)90099-4.
- [28] DANIEL LAMBERTZ u. a. „A Reproducibility Study on Musculotendinous Stiffness Quantification, Using a New Transportable Ankle Ergometer Device“. In: *Journal of biomechanics* 41.15 ((Jan. 2008)). Evaluation Study Journal Article Research Support, Non-U.S. Gov’t, S. 3270–3. ISSN: 0021-9290. DOI: 10.1016/j.jbiomech.2008.08.022.
- [29] A. L. HOF. „In Vivo Measurement of the Series Elasticity Release Curve of Human Triceps Surae Muscle“. In: *Journal of biomechanics* 31.9 ((Jan. 1998)). Journal Article, S. 793–800. ISSN: 0021-9290. DOI: 10.1016/s0021-9290(98)00062-1.
- [30] M. ZEE und M. VOIGT. „Moment Dependency of the Series Elastic Stiffness in the Human Plantar Flexors Measured in Vivo“. In: *Journal of biomechanics* 34.11 ((Jan. 2001)). Journal Article Research Support, Non-U.S. Gov’t, S. 1399–406. ISSN: 0021-9290. DOI: 10.1016/s0021-9290(01)00133-6.
- [31] P. L. WEISS, I. W. HUNTER und R. E. KEARNEY. „Human Ankle Joint Stiffness over the Full Range of Muscle Activation Levels“. In: *Journal of biomechanics* 21.7 ((Jan. 1988)). Journal Article Research Support, Non-U.S. Gov’t PII: 0021929088902175, S. 539–44. ISSN: 0021-9290. DOI: 10.1016/0021-9290(88)90217-5.
- [32] DAN EXETER und DAVID A. CONNELL. „Skeletal Muscle: Functional Anatomy and Pathophysiology“. eng. In: *Seminars in Musculoskeletal Radiology* 14.2 ((Juni 2010)). Journal Article Review, S. 97–105. ISSN: 1098-898X. DOI: 10.1055/s-0030-1253154.
- [33] ZOHREH BARANI LONBANI. „MICRO-CT BASED CHARACTERISATION OF CHANGES TO THE VASCULAR NETWORK FOLLOWING CLOSED SOFT TISSUE TRAUMA AND CRYOTHERAPY“. en. In: (), S. 248.
- [34] LUDOVIC GAUT und DELPHINE DUPREZ. „Tendon Development and Diseases: Tendon Development and Diseases“. en. In: *Wiley Interdisciplinary Reviews: Developmental Biology* 5.1 ((Jan. 2016)), S. 5–23. ISSN: 17597684. DOI: 10.1002/wdev.201.
- [35] JAMES H.-C. WANG. „Mechanobiology of Tendon“. en. In: *Journal of Biomechanics* 39.9 ((Jan. 2006)). Journal Article Research Support, N.I.H., Extramural Research Support, Non-U.S. Gov’t Review, S. 1563–1582. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2005.05.011.
- [36] MIQUEL DALMAU-PASTOR u. a. „Anatomy of the Triceps Surae: A Pictorial Essay“. eng. In: *Foot and Ankle Clinics* 19.4 ((Dez. 2014)). Journal Article Review, S. 603–635. ISSN: 1558-1934. DOI: 10.1016/j.fcl.2014.08.002.

- [37] R. L. LIEBER u. a. „Frog Semitendinosus Tendon Load-Strain and Stress-Strain Properties during Passive Loading“. In: *American Journal of Physiology-Cell Physiology* 261.1 ((Juli 1991)), S. C86–C92. ISSN: 0363-6143. DOI: 10.1152/ajpcell.1991.261.1.C86.
- [38] G. F. ELLIOTT, J. LOWY und C. R. WORTHINGTON. „An X-Ray and Light-Diffraction Study of the Filament Lattice of Striated Muscle in the Living State and in Rigor“. en. In: *Journal of Molecular Biology* 6.4 ((Apr. 1963)), 295–IN9. ISSN: 0022-2836. DOI: 10.1016/S0022-2836(63)80090-X.
- [39] ICHINO MATSUBARA und GERALD F. ELLIOTT. „X-Ray Diffraction Studies on Skinned Single Fibres of Frog Skeletal Muscle“. en. In: *Journal of Molecular Biology* 72.3 ((Dez. 1972)), S. 657–669. ISSN: 0022-2836. DOI: 10.1016/0022-2836(72)90183-0.
- [40] PETER A. HUIJING. „Muscle, the Motor of Movement: Properties in Function, Experiment and Modelling“. en. In: *Journal of Electromyography and Kinesiology* 8.2 ((Apr. 1998)), S. 61–77. ISSN: 1050-6411. DOI: 10.1016/S1050-6411(97)00023-0.
- [41] CHRISTOPHER J. ARELLANO u. a. „Determinants of Aponeurosis Shape Change during Muscle Contraction“. en. In: *Journal of Biomechanics* 49.9 ((Juni 2016)), S. 1812–1817. ISSN: 0021-9290. DOI: 10.1016/j.jbiomech.2016.04.022.
- [42] M. ARATOW u. a. „Intramuscular Pressure and Electromyography as Indexes of Force during Isokinetic Exercise“. In: *Journal of Applied Physiology* 74.6 ((Juni 1993)), S. 2634–2640. ISSN: 8750-7587. DOI: 10.1152/jappl.1993.74.6.2634.
- [43] DAVID SLEBODA und THOMAS ROBERTS. „Incompressible Fluid Plays a Mechanical Role in the Development of Passive Muscle Tension“. In: *Biology Letters* 13 ((Jan. 2017)), S. 20160630. DOI: 10.1098/rsbl.2016.0630.
- [44] E. AZIZI u. a. „Resistance to Radial Expansion Limits Muscle Strain and Work“. en. In: *Biomechanics and Modeling in Mechanobiology* 16.5 ((Okt. 2017)), S. 1633–1643. ISSN: 1617-7940. DOI: 10.1007/s10237-017-0909-3.
- [45] PETER A HUIJING. „Muscle as a Collagen Fiber Reinforced Composite: A Review of Force Transmission in Muscle and Whole Limb“. en. In: *Journal of Biomechanics* 32.4 ((Apr. 1999)), S. 329–345. ISSN: 0021-9290. DOI: 10.1016/S0021-9290(98)00186-9.
- [46] HUUB MAAS und TAIJA FINNI. „Mechanical Coupling Between Muscle-Tendon Units Reduces Peak Stresses“. eng. In: *Exercise and Sport Sciences Reviews* 46.1 ((Jan. 2018)), S. 26–33. ISSN: 1538-3008. DOI: 10.1249/JES.0000000000000132.
- [47] EMANUEL AZIZI und THOMAS J. ROBERTS. „Biaxial Strain and Variable Stiffness in Aponeuroses“. en. In: *The Journal of Physiology* 587.17 ((2009)), S. 4309–4318. ISSN: 1469-7793. DOI: 10.1113/jphysiol.2009.173690.

- [48] MARCELO EPSTEIN, MAX WONG und WALTER HERZOG. „Should Tendon and Aponeurosis Be Considered in Series?“ eng. In: *Journal of Biomechanics* 39.11 ((2006)), S. 2020–2025. ISSN: 0021-9290. DOI: 10.1016/j.jbiomech.2005.06.011.
- [49] DANIEL BODOR. „Not the Plantaris – Keys to Better Diagnosis of Calf Strain Injuries“. <https://radsourc.us/not-plantaris-keys-better-diagnosis-calf-strain-injuries/>. (Dez. 2020).
- [50] C. BOSCO u. a. „Store and Recoil of Elastic Energy in Slow and Fast Types of Human Skeletal Muscles“. In: *Acta physiologica Scandinavica* 116.4 ((Jan. 1982)). Journal Article Research Support, Non-U.S. Gov’t, S. 343–9. ISSN: 0001-6772. DOI: 10.1111/j.1748-1716.1982.tb07152.x.
- [51] G. A. LICHTWARK und A. M. WILSON. „A Modified Hill Muscle Model That Predicts Muscle Power Output and Efficiency during Sinusoidal Length Changes“. In: *The Journal of experimental biology* 208.Pt 15 ((Jan. 2005)). Comparative Study Journal Article Research Support, Non-U.S. Gov’t, S. 2831–43. DOI: 10.1242/jeb.01709.
- [52] MARK L. LATASH und VLADIMIR M. ZATSIORSKY. „Joint Stiffness: Myth or Reality?“ en. In: *Human Movement Science* 12.6 ((Dez. 1993)), S. 653–692. ISSN: 01679457. DOI: 10.1016/0167-9457(93)90010-M.
- [53] MARTYN R. SHORTEN. „Muscle Elasticity and Human Performance“. en. In: *Medicine and Sport Science*. Hrsg. von B. VAN GHELUWE und J. ATHA. Bd. 25. S. Karger AG, (Jan. 1988), S. 1–18. ISBN: 978-3-8055-4546-4 978-3-318-03939-9. DOI: 10.1159/000414393.
- [54] G. C. AGARWAL und G. L. GOTTLIEB. „Oscillation of the Human Ankle Joint in Response to Applied Sinusoidal Torque on the Foot“. In: *The Journal of physiology* 268.1 ((Jan. 1977)). Journal Article Research Support, U.S. Gov’t, Non-P.H.S. Research Support, U.S. Gov’t, P.H.S. Journal Article Research Support, U.S. Gov’t, Non-P.H.S. Research Support, U.S. Gov’t, P.H.S., S. 151–76. ISSN: 0022-3751. DOI: 10.1113/jphysiol.1977.sp011852.
- [55] C. N. MAGANARIS und J. P. PAUL. „In Vivo Human Tendon Mechanical Properties“. In: *The Journal of physiology* 521 Pt 1 ((Jan. 1999)). Journal Article, S. 307–13. ISSN: 0022-3751. DOI: 10.1111/j.1469-7793.1999.00307.x.
- [56] A. V. HILL. „The Heat of Shortening and the Dynamic Constants of Muscle“. In: ((Jan. 1938)). DOI: 10.1098/rspb.1938.0050.
- [57] S BOUISSET und F GOUBEL. „Integrated Electromyographical Activity and Muscle Work.“ en. In: *Journal of Applied Physiology* 35.5 ((Nov. 1973)), S. 695–702. ISSN: 8750-7587, 1522-1601. DOI: 10.1152/jappl.1973.35.5.695.

- [58] CHRISTOPHE CORNU und FRANCIS GOUBEL. „Musculo-Tendinous and Joint Elastic Characteristics during Elbow Flexion in Children“. en. In: *Clinical Biomechanics* 16.9 ((Nov. 2001)), S. 758–764. ISSN: 02680033. DOI: 10.1016/S0268-0033(01)00076-6.
- [59] G. J. ETTEMA und P. A. HUIJING. „Skeletal Muscle Stiffness in Static and Dynamic Contractions“. In: *Journal of biomechanics* 27.11 ((Jan. 1994)). Comparative Study Journal Article, S. 1361–8. ISSN: 0021-9290. DOI: 10.1016/0021-9290(94)90045-0.
- [60] M. I. ALMEIDA-SILVEIRA u. a. „Changes in Stiffness Induced by Hindlimb Suspension in Rat Achilles Tendon“. In: *European journal of applied physiology* 81.3 ((Jan. 2000)). Journal Article, S. 252–7. ISSN: 1439-6319. DOI: 10.1007/s004210050039.
- [61] „Long-Term Bed Rest Study 2019 - Artificial Gravity Bed Rest Study: A Joint Study between the National Aeronautics and Space Administration, the European Space Agency and the German Aerospace Center“. In: ((Apr. 2018)).
- [62] PAOLO BARATTINI, SUZANNE SCHNEIDER und V REGGIE EDGERTON. „MARES: A NEW TOOL FOR MUSCULAR, NEUROMUSCULAR AND EXERCISE RESEARCH IN THE INTERNATIONAL SPACE STATION“. en. In: 12 ((2005)), S. 8.
- [63] F NÚÑEZ u. a. „BODY POSITION REPRODUCIBILITY AND JOINT ALIGNMENT STABILITY CRITICALITY ON A MUSCULAR STRENGTH RESEARCH DEVICE“. en. In: (), S. 2.
- [64] SENER AEROSPACIAL. „Muscle Atrophy Research and Exercise System (MARES): <https://www.aerospacial.sener/en/products/muscle-atrophy-research-exercise-system-mares>“. In: ().
- [65] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. „STS-131 Experiment Express“. https://www.nasa.gov/pdf/435885main_sts131_press_kit.pdf. (2010).
- [66] KIRK ENGLISH u. a. „A Ground-Based Comparison of the Muscle Atrophy Research and Exercise System (MARES) and a Commercially Available Isokinetic Dynamometer“. In: *Acta Astronautica* 92 ((Nov. 2013)), S. 3–9. DOI: 10.1016/j.actaastro.2012.06.015.
- [67] „L+129: Logbuch — Logbuch ISS“. <https://www.logbuch-iss.de/2015/04/02/l129-logbuch/>.
- [68] JORDI ELVIRA u. a. „Evaluation of Anthropometric Requirements for the Design of an Ergometer Restraint System“. en. In: *31st International Conference On Environmental Systems*. (Juli 2001), S. 2001–01–2186. DOI: 10.4271/2001-01-2186.
- [69] EUROPEAN SPACE AGENCY. „MARES_description_v2.Pdf“. http://wsn.spaceflight.esa.int/docs/others/MARES_description_v2.pdf.
- [70] „ESA - Muscle Atrophy Research and Exercise System“. https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Columbus/Muscle_Atrophy.

- [71] R. E. KEARNEY und I. W. HUNTER. „System Identification of Human Joint Dynamics“. eng. In: *Critical Reviews in Biomedical Engineering* 18.1 ((1990)). Journal Article Review, S. 55–87. ISSN: 0278-940X.
- [72] LLOYD B. ELDRED, WILLIAM P. BAKER und ANTHONY N. PALAZOTTO. „Kelvin-Voigt versus Fractional Derivative Model as Constitutive Relations for Viscoelastic Materials“. en. In: *AIAA Journal* 33.3 ((März 1995)), S. 547–550. ISSN: 0001-1452, 1533-385X. DOI: 10.2514/3.12471.
- [73] MOON JEONG KANG und HONG HEE YOO. „In Vivo Viscoelastic Properties of Human Thigh under Compression Estimated by Experimental Results Obtained with Pendulum Test“. en. In: *International Journal of Precision Engineering and Manufacturing* 18.9 ((Sep. 2017)), S. 1253–1262. ISSN: 2234-7593, 2005-4602. DOI: 10.1007/s12541-017-0147-8.
- [74] GREGORY W OJAKANGAS. „Dynamics of a Low Mass, Friction Reduced Planar Arm Model with Servo Regulated Viscoelastic Muscles in a Microgravity Environment DLMFRPAMSVMMME“. en. In: (), S. 32.
- [75] „Praktische Bestimmung Des Trägheitsmomentes Eines Starren Körpers“. <https://www.paehler.org/tim/archiv/extern/david/htmlexamen/22Bestimmung.html>.
- [76] PAR F. GOUBEL und E. PERTUZON. „Évaluation De L Élasticité Du Muscle in Situ Par Une Méthode De Quick-Release“. In: *Archives internationales de physiologie et de biochimie* 81.4 ((Jan. 1973)). Journal Article, S. 697–707. ISSN: 0003-9799. DOI: 10.3109/13813457309074474.
- [77] R. W. ANGEL, W. EPPLER und A. IANNONE. „Silent Period Produced by Unloading of Muscle during Voluntary Contraction“. In: *The Journal of physiology* 180.4 ((Jan. 1965)). Journal Article, S. 864–70. ISSN: 0022-3751. DOI: 10.1113/jphysiol.1965.sp007736.
- [78] DANIEL LAMBERTZ u. a. „Evaluation of Musculotendinous Stiffness in Prepubertal Children and Adults, Taking into Account Muscle Activity“. In: *Journal of applied physiology (Bethesda, Md. : 1985)* 95.1 ((Jan. 2003)). Clinical Trial Comparative Study Journal Article Research Support, Non-U.S. Gov’t, S. 64–72. ISSN: 8750-7587. DOI: 10.1152/japplphysiol.00885.2002.
- [79] I. W. HUNTER und R. E. KEARNEY. „Dynamics of Human Ankle Stiffness: Variation with Mean Ankle Torque“. In: *Journal of biomechanics* 15.10 ((Jan. 1982)). Journal Article Research Support, Non-U.S. Gov’t, S. 747–52. ISSN: 0021-9290. DOI: 10.1016/0021-9290(82)90089-6.
- [80] „Resonant Frequency from Bode Plot - Electrical Engineering Stack Exchange“. <https://electronics.stackexchange.com/questions/112521/resonant-frequency-from-bode-plot>.

Anhang A

Anhang

A.1 Technische Spezifikationen des MARES

Die folgenden Spezifikationen sind der Website der ESA entnommen. [70]

Torque and angular position/velocity measurements and training on joint movements, both left and right:

- Knee flexion/extension
- Ankle flexion/extension
- Trunk flexion/extension
- Hip flexion/extension
- Shoulder flexion/extension
- Elbow flexion/extension
- Wrist flexion/extension; supination/pronation; radial/ulnar deviation

Force and linear position and velocity measurements and training are supported on the following multi-joint movements:

- Whole arm linear press (front, overhead and intermediate trajectories)
- Whole leg linear press (front, down and intermediate trajectories)

MARES supports exercise motions in these modes:

- Isometric
- Isokinetic (concentric and eccentric)

- Isotonic (concentric and eccentric)
- Simulation of ideal elements: spring, friction and inertia
- Parameter control following a predefined pattern: position control, velocity control, torque/force control, power control
- Quick release of free motion

Complex combinations of these modes are possible.

Maximum ratings:

- Torque (Nm): ± 450 continuous, ± 900 peak (200 ms)
- Force (N): ± 250
- Angular velocity (rad/s): ± 9 ($515^\circ/\text{s}$) concentric and eccentric
- Linear velocity (m/s): ± 0.5
- Mechanical power (W): 2700 continuous, 4500 peak

Accuracy:

- Torque: ± 0.3 Nm for low torques, $\pm 0.5\%$ for high torques; 500 Hz
- Force: ± 0.125 N; 500 Hz
- Angular velocity: $\pm 0.2^\circ/\text{s}$ for low velocities, $\pm 0.5\%$ for high velocities; 200 Hz
- Linear velocity: ± 1 mm/s for low velocities, $\pm 0.1\%$ for high velocities; 200 Hz
- Angular position: $\pm 0.5^\circ$; 200 Hz
- Linear position: ± 0.5 mm; 200 Hz

Size and power consumption:

Mass: 200 kg Main box size: 95 x 45 x 45 cm Power: 300 W average, 1000 W peak

- Mass: 200 kg
- Main box size: 95 x 45 x 45 cm
- Power: 300 W average, 1000 W peak

A.2 Steifigkeiten zu alten Endstoppositionen

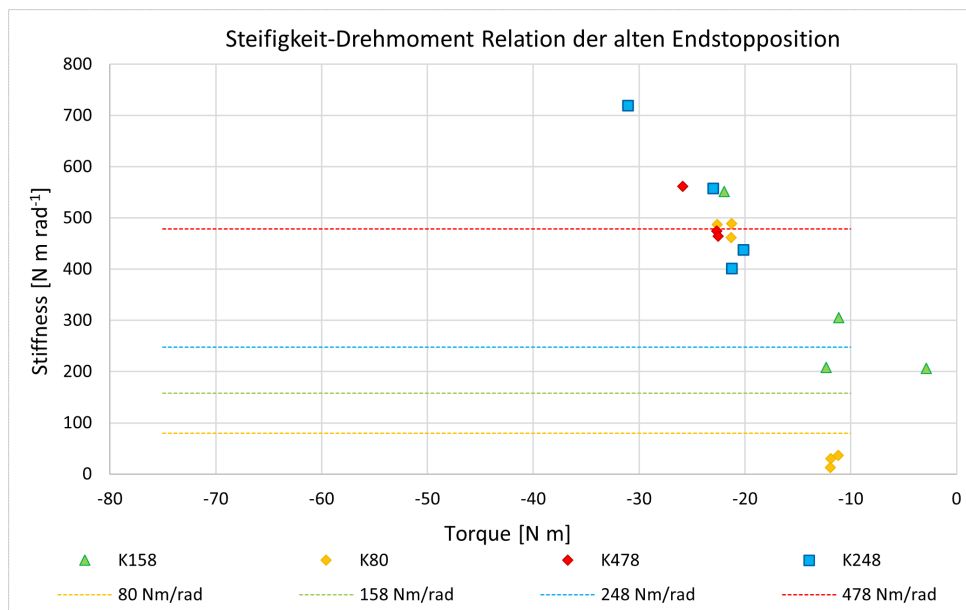


Abbildung A.1: Darstellung der Relation von ermittelter Steifigkeit und angelegtem Drehmoment der Federn K_{478} (rot), K_{158} (grün), K_{80} (gelb) und sowie K_{248} (blau) bei engeren, später verworfenen Positionen der Endstops

A.3 Verwendeter Code

A.3.1 QR

A.3.1.1 *proc_MARES_mv_v0.2.r*

```

1  ##### AGBRESA study (enviHab, 2019), plot raw data (test.r), spline,
                                     derivative, stiffness #####
2  versionNo = "0.2";
3  # Author: M.V.; U.M.
4
5  ##working directory##
6  setwd( "C:/Users/Milan Vincentz/Documents/Studium/Masterarbeit/
                                     MARES_Auswertung/
                                     AGBRESA_Outcome_Sample" );
7  library(signal);
8
9
10 ritMARES.ReadChannel = function( path, strChannel ) {
11   fN = paste( path, strChannel, sep="/" );
12   kBytes = file.info(fN)$size/1024 + 100;
13   return(readBin( con=fN, what="double", n=kBytes*256, size=8, endian="big"
14                  ));

```



```

15
16 ritMARES.UpdateTorque = function( path, frqLoPass=20 ) {
17   trq = -ritMARES.ReadChannel( path, "Subject_Torque" );
18   bInflight=grep( pattern="INFLIGHT", x=path, ignore.case=T );
19   if( length(bInflight) ) trq = 1.38 + 1.01003*trq;
20   if( frqLoPass==0 ) return(trq);
21   BW.filter = butter( 4, frqLoPass/(smpRateMARES/2), "low" );
22   d.trq <- rev( filter( filt=BW.filter, rev( filter( filt=BW.filter, x=trq
23                                     )))); # filtered signal
24
25 ritMARES.UpdateTM1 = function( path ) {
26   d.TM1 <- ritMARES.ReadChannel( path, "Time_Mark_1" );
27 }
28
29 ritMARES.UpdateData = function( path ) {
30   # if( path != ritMAREScurPath ) {
31   #   ritMAREScurPath <- path;
32   #   ritMARES.UpdateTM1(path);
33   #   N.MARES.Smp <- length(d.Trq);
34   #   ritMARES.UpdateTorque( path );
35
36   d.trq <- ritMARES.ReadChannel( path, "Subject_Torque" );
37
38   ## Einlesen der Datenstreams ##
39   d.Ang <- ritMARES.ReadChannel( path, "Angle" );
40   d.Pos <- ritMARES.ReadChannel( path, "Angular_Position" );
41   d.Vel <- ritMARES.ReadChannel( path, "Angular_Velocity" );
42   d.EMG1 <- ritMARES.ReadChannel( path, "Ext_Channel_1" );
43   d.EMG2 <- ritMARES.ReadChannel( path, "Ext_Channel_2" );
44   d.EMG3 <- ritMARES.ReadChannel( path, "Ext_Channel_3" );
45   d.EMG4 <- ritMARES.ReadChannel( path, "Ext_Channel_4" );
46   if( file.exists( paste0( path, "/Ext_Channel_6" ) ) ) {
47     d.Gon <- ritMARES.ReadChannel( path, "Ext_Channel_6" );
48   } else d.Gon <- NULL;
49   d.TM1 <- ritMARES.ReadChannel( path, "Time_Mark_1" );
50   # }
51 }
52
53 ritMARES.UpdateData( "C:/Users/Milan Vincentz/Documents/Studium/Masterarbeit
54                               /MARES_Auswertung/
55                               AGBRESA_Outcome_Sample/1 Data/Raw/
56                               MARES/ANKLE2_G/temporal_data/
57                               Testreihen SPQR/neu/10August2020[
58                               08h48m31s]" );
59
60 #### read txt files ####

```

```

56 # txt=read.csv2("C:/Users/vinc_mi/Documents/Masterarbeit_DLR/
                                     MARES_Auswertung/Export/test.txt",
                                     sep="\t")
57
58 dl=length(d.Ang)
59 x=seq(from = 1, to = dl, by = 1)
60
61 ##### QRs #####
62
63 # SPQR specific QR #
64 Posspqr <- d.Pos[0:1500000] # Analysebereich Positionsstream
65 Trqspqr <- d.trq[0:1500000] # Analysebereich Drehmomentsstream
66 xspqr <- x[0:1500000] # Analysebereich x-Achse
67 velspqr <- d.Vel[0:1500000] # Analysebereich Winkelgeschwindigkeit
68 xms=xspqr/2 # Umrechnung von Skalenstrichen in Millisekunden
69
70 # # All QRs subject B #
71 # PosQR <- d.Pos[5700000:11250000]
72 # TrqQR <- d.trq[5700000:11250000]
73 # xQR <- x[5700000:11250000]
74
75 ##### spline #####
76 xseq=xspqr/2000 # Umrechnung in Sekunden
77 yseq=Posspqr
78 posspl=splinefun(xseq, yseq) # Splinekurve ueber Stream
79 ## 1. und 2. Ableitung ##
80 posderiv1=posspl(xseq,deriv=1) #1. Ableitung = Geschwindigkeit
81 posderiv2=posspl(xseq,deriv=2) #2. Ableitung = Beschleunigung
82
83 ##### filter #####
84 bf1 <- butter(4, 0.2, type = c("pass"))
85 bf2 <- butter(4, 0.05, type = c("pass"))
86 filtposderiv1 <- filter(bf1, posderiv1)
87 filtposderiv2 <- filter(bf2, posderiv2)
88
89 ##### plot #####
90 par(mfrow=c(2,1)) # Anzahl der ausgegebenen Zeilen und Spalten
91 ## one QR ##
92 plot(xms, Posspqr, ylim=c(174, 184), lwd=1, type="l", main = "10August2020
                                     [08h48m31s]", xlab = "Zeit [ms]",
                                     ylab = "Angular_Position [deg]")
93 plot(xms, Trqspqr, ylim=c(-120, 62), lwd=1, type="l", xlab = "Zeit [ms]",
                                     ylab = "Subject_Torque [Nm]")
94 # plot(xseq, filtposderiv1, type = "l", xlab = "time [s]", ylab = "vel [deg/s
                                     ]")
95 # plot(xseq, filtposderiv2, type = "l", xlab = "time [s]", ylab = "acc [deg/s
                                     ^2]")

```

```

96 # plot(xseq, velspqr, lwd=1, type="l", xlab = "time [s]", ylab = "vel MARES
    [deg/s^2]" )
97
98 ## all QR ##
99 # plot(x, d.Pos, ylim=c(165, 185), lwd=1, type="l", main = "QR subject ",
    xlab = "time [s]", ylab = "AngPos [
    deg]")
100 # plot(x, d.trq, ylim=c(-50, 80), lwd=1, type="l", xlab = "time [s]", ylab =
    "Torque [Nm]")
101
102 ##### stiffness #####
103 Pos0ms <- as.numeric(as.character(d.Pos[9733836])) #Winkelposition Start
    Intervall
104 Pos20ms <- as.numeric(as.character(d.Pos[9733876])) #Winkelposition Ende
    Intervall
105 deltaposderiv2 <- filtposderiv2[40]-filtposderiv2[1] # Delta Beschleunigung
106 deltapos <- Pos20ms-Pos0ms #Delta Position
107 posderiv2max <- max(filtposderiv2[1:40]) # Maximale Beschleunigung
108
109 S = deltaposderiv2/deltapos*-mean(Trq1)/posderiv2max #Steifigkeit fuer
    ausgewaehltes Intervall

```

A.3.1.2 mares_QRanalysis4 – 0.py

```

1 # -*- coding: utf-8 -*-
2 """
3 First created on Tue Mar 24 13:03:13 2020
4 version from 17.12.2020
5
6 @author: mittag
7 """
8 import struct
9 from matplotlib import pyplot as plt
10 import numpy as np
11 from scipy.signal import butter, filtfilt
12 from datetime import datetime as dt
13 from datetime import timedelta as td
14 import os
15 from scipy.optimize import curve_fit
16
17 def lowpass(data, mrange): #low pass filter as running average
18     y=1*data
19     for i in range(mrange, len(data)-mrange):
20         y[i]=np.mean(data[i-mrange:i+mrange])
21     return y
22
23 def butter_lowpass_filter(data, cutoff, fs, order):

```

```

24     normal_cutoff = cutoff / nyq
25     # Get the filter coefficients
26     b, a = butter(order, normal_cutoff, btype='lowpass', analog=False)
27     y = filtfilt(b, a, data)
28     return y
29
30 def cosfit(x, g, w, c, dx):
31     return g * np.cos(w*(x+dx))+c
32 def sinfit(x, g, w, c, dx):
33     return g * np.sin(w*(x+dx))+c
34
35
36 def calcR2(val, fit): # calculation of coefficient of determination for fir
                        # results
37     global QR; global mm
38     mm=fit
39     SQR=0
40     SQT=0
41     QR=val-fit
42     valmean=np.mean(val)
43     for i in range(0,len(val)):
44         SQR=SQR+(val[i]-fit[i])**2 # variance
45         SQT=SQT+(val[i]-valmean)**2 # residual sum of squares
46     R2=1-SQR/SQT
47     return R2
48
49
50 # prepare result directoy using time tag
51 ct = dt.now()
52 ttag=str(ct.year)+'-'+str(ct.month)+'-'+str(ct.day)+'_'+str(ct.hour)+'-'+str
                        (ct.minute)
53 sdir='results/'+ttag
54 if not os.path.exists(sdir):
55     os.makedirs(sdir)
56
57 # prepare result file, write header
58 ivdata = open(sdir+"/MaresTableQR.csv",'w');
59 ivdata.write("SubjID;IntervalStartDateTime(UTC);IntervalStartIndex;
                        IntervalEndIndex;IntervalMeanTorque
                        [Nm];IntervalMaxAngAcceleration[rad
                        /sec2];IntervalMomentOfInertia[kgm2
                        ];IntervalStiffness[Nm/rad);
                        IntvStiffnSTD;MoIfit[kgm2]; Sfit[Nm
                        /rad]; R2; File\n")
60 ivdata.close()
61
62 # scan rate of input data
63 fs=2000

```

```

64 nyq=0.5*fs
65 err=-999
66
67 tmlvl=110 # analyse QR intervalls
68
69 #----- parse data files in base directory
70 basedirMares="Y:/Pro/AGBRESA1/PI/Rittweger/MARES/dekodiert/Agbresa 1 Ankle 2
      "
71 sesss=[]
72 listOfSubjs = os.listdir(basedirMares)
73 for Subj in listOfSubjs:
74     listOfSesss = os.listdir(basedirMares+"/"+Subj)
75     for Sess in listOfSesss:
76         # get time stamp from session folder name
77         dtim=dt.strptime(Sess,'%d%B%Y[%Hh%Mm%Ss]')
78         session=basedirMares+"/"+Subj+"/"+Sess
79         print(session)
80         # read data streams, check availability and consistency
81         try :
82             data=open(session+"/Time_Mark_1","rb").read()
83             trq0=open(session+"/Subject_Torque","rb").read()
84             pos0=open(session+"/Angular_Position","rb").read()
85             vel0=open(session+"/Angular_Velocity","rb").read()
86         except :
87             print("")
88             print("missing data")
89         else :
90             ld=len(data)
91             if ((ld!=len(trq0)) or (ld!=len(pos0)) or (ld!=len(vel0)) ) :
92                 print("corrupted data")
93             else :
94
95
96                 tm1=[]; trq=[]; pos=[]; vel=[]
97                 # convert binary data to double with order big endian
98                 for i in range(0,ld,8):
99                     rx = struct.unpack('>d',data[i:i+8])[0]; tm1.append(rx) #
100                                     Time_Mark_1
101                     rx = struct.unpack('>d',trq0[i:i+8])[0]; trq.append(rx) #
102                                     Subject_Torque
103                     rx = struct.unpack('>d',pos0[i:i+8])[0]; pos.append(rx) #
104                                     Angular_Position
105                     rx = struct.unpack('>d',vel0[i:i+8])[0]; vel.append(rx) #
106                                     Angular_Velocity
107
108                 trq=np.array(trq)
109                 trq=butter_lowpass_filter(trq, 20, fs, 4) # low pass filter
110                                     torque stream

```

```

106
107         dl=len(tm1) # size of total stream
108
109         # identify QR intervals
110         na=[]; ne=[]; x=[]
111         x.append(0)
112         for i in range(1,dl): # find QR intervals
113             if (tm1[i] - tm1[i-1]>tmlvl-5 and tm1[i] - tm1[i-1]<tmlvl+5)
114                 :
115                 na.append(i)
116                 if (tm1[i] - tm1[i-1]> -tmlvl-5 and tm1[i] - tm1[i-1]< -
117                     tmlvl+5) :
118
119                     ne.append(i)
120                 x.append(i) # generation of "time"-stream, every step is 0.5
121                             ms
122
123         lna=len(na) # number of QR intervals found
124         if lna<=0:
125             print("no QR intervals found for this session")
126         else :
127             if (lna-len(ne))==1 :
128                 ne.append(na[lna-1]+30000)
129                 if ne[lna-1]>len(trq):
130                     na.remove(na[lna-1])
131                     ne.remove(ne[lna-1])
132                     lna=lna-1
133
134             icnt=0
135             for nisp in range(0,lna): # analyse all intervals
136                 icnt=icnt+1
137                 cnt=1
138                 velspan=-min(vel[na[nisp]:ne[nisp]]) # delta to QR (
139                                                         negative)
140                                                         maximum
141
142                 for i in range(na[nisp]+2, ne[nisp]) :
143                     cnt=cnt+1
144                     if vel[i]<vel[na[nisp]]-velspan/2 :
145                         qrnw=cnt # where down-jump is half-way set "
146                                     qrnw"
147
148                         break
149
150                 owin=-300 # offset of plotwindow with respect to qrnw
151                 o0=-40 # offset of estimated 0-point wrt qrnw
152                 ofita=-60 # offset of fit window start wrt qrnw
153                 #ofite=+38 # offset of fit window end wrt qrnw
154                 ofite=+110 # offset of fit window end wrt qrnw
155                 otmeana=-200 #offset of mean torque calc start wrt qrnw
156                 otmeane=-100 #offset of mean torque calc end wrt qrnw

```

```

148
149     # offsets of analysis window around QR event
150     qrnowa=qrnow+na[nisp]+owin
151     qrnowe=qrnow+na[nisp]+150
152
153     # estimated offset of 0-point for 1st fit
154     qr0est=qrnow+na[nisp]+o0
155
156     # offsets of fit windows relative to analysis window
157     ifs=qrnow+na[nisp]+ofita-qrnowa
158     ife=qrnow+na[nisp]+ofite-qrnowa
159
160     # define vectors around QR event for plots
161     xsp= np.arange(qrnowa, qrnowe)
162     tm1sp=tm1[qrnowa: qrnowe]
163     trqsp=trq[qrnowa: qrnowe]
164     possp=pos[qrnowa: qrnowe]
165     possp=np.array(possps)
166     velsp=vel[qrnowa: qrnowe]
167     velsp=np.array(velsp)
168     velsp2=1*velsp
169     accsp2=1*velsp
170     possprad=1*possps/180*np.pi
171     velsp2=1*velsp/180*np.pi
172
173     # vectors for fit plots
174     xf=xsp[ifs:ife]-qr0est
175
176     if possprad[3]<0 :
177         possprad=possprad+360;
178     posf=possprad[ifs:ife]
179     velf=velsp2[ifs:ife]
180
181
182
183     # differentiation of angular position
184     for i in range(1, len(possprad)):
185         velsp2[i]=(possprad[i]-possprad[i-1])*2000
186     velsp2[0]=0
187     velsp2unf=1*velsp2
188     #velsp2=butter_lowpass_filter(velsp2, 10, fs, 4)
189     velsp2=lowpass(velsp2,3)
190     velsp2=lowpass(velsp2,3)
191     velf2=velsp2[ifs:ife]
192     QRstart=0
193     for i in range(0, len(velf2)):
194         if abs(velf2[i])>0.2:
195             QRstart=i-4

```

```

196         break
197     if QRstart<0:
198         QRstart=0
199     velf2unf=velsp2unf[ifs:ife]
200     # differentiation of angular velocity
201     for i in range(2, len(possprad)-1):
202         accsp2[i]=(velsp2[i]-velsp2[i-1])*2000
203     accsp2[0]=0
204     accsp2[1]=0
205     accsp2unf=1*accsp2
206     # double low pass filtering of angular acceleration
207     accsp2=lowpass(accsp2,5)
208     accsp2=lowpass(accsp2,5)
209     accf2=accsp2[ifs:ife]
210     accf2unf=accsp2unf[ifs:ife]
211     amax=min(accf2)
212     amaxpos=np.argmax(accf2)
213     T0=np.mean(trqsp[ifs+otmeana:ifs+otmeane]) # calculate
                                                T0 as mean over
                                                "torque window
                                                "
214
215     wmin=amaxpos+2 # put start of analysis window just
                                                behind position
                                                of (negative)
                                                maximum of
                                                acceleration
216     if wmin>55 : # in case of late maximum put wmin to step
                                                55
217         wmin=55
218     wmax= wmin+15 #only used for plot below
219     Q0=[]
220     for i in range(wmin,wmin+21,3): # multiple computation
                                                od
                                                delta_acceleration
                                                /delta_position
                                                ; delta over 15
                                                steps
221         q=(accf2[i+15]-accf2[i])/(posf[i+15]-posf[i])
222         Q0.append(q)
223     Q0=np.array(Q0)
224     Q=np.mean(Q0)
225     Qstd=np.std(Q0)
226     # calculation of moment of inertia (I) and stiffness (S)
227     I=T0/(amax)
228     S=-Q*I
229     Sstd=Qstd*I
230

```



```

231         #alternate calculation of I and S using cosine-fit to
                                                acceleration
                                                curve for
                                                smoothening

232         fitmin=wmin;
233         fitmax=wmin+36
234         fitxf=xf[fitmin:fitmax]
235         fita=accf2[fitmin:fitmax]
236         ps0=possp[5]
237         sgn=1
238         if amax<0:
239             sgn=-1
240             fita=fita*sgn
241         popta, pcova = curve_fit(cosfit, fitxf, fita, bounds=(
                                                .6*sgn*amax,0,-
                                                0.001,-QRstart+
                                                19], [2*sgn*
                                                amax, 0.1, 0.
                                                001, -QRstart+
                                                20]))

242         print(sgn*amax,popta)
243         # calculate quality of fit as coefficient of
                                                determination
                                                R2

244         fitf=cosfit(xf,*popta)
245         fitR=cosfit(fitxf,*popta)
246         R2=calcR2(fita,fitR)
247         print(R2)
248
249         Qcosfit=(fitf[wmin+15]-fitf[wmin])/(posf[wmin+15]-posf[
                                                wmin]) # single
                                                calculation of

                                                delta_acceleration
                                                /delta_position

250         Icosfit=-T0/popta[0]
251         Scosfit=Qcosfit*Icosfit
252
253         # generate overview plots for assessment of quality
254         xmin= min(xf)
255         xmax= max(xf)
256         fig = plt.figure()
257         ax = plt.axes(xlim=(xmin,xmax), ylim=(-60,30))
258         ax.set_xlabel("time [0.5ms]")
259         #ax.set_ylabel('TM1')
260         psline, = ax.plot([], [],lw=1, color='cyan')
261         psline.set_data(xf, (posf-possprad[3])*200)
262         vlone, = ax.plot([], [],lw=1, color='blue')

```

```

263         velfm=(velf-velsprad[4])*5
264         vline.set_data(xf, velfm)
265
266         v2uline, = ax.plot([], [],lw=1, color='gray')
267         velf2um=(velf2unf-velsp2unf[4])*5
268         v2uline.set_data(xf, velf2um)
269
270         v2line, = ax.plot([], [],lw=1, color='black')
271         velf2m=(velf2-velsp2[4])*5
272         v2line.set_data(xf, velf2m)
273
274         auline, = ax.plot([], [],lw=1, color='green')
275         accf2um=accf2unf/10
276         auline.set_data(xf, accf2um)
277
278         aline, = ax.plot([], [],lw=1, color='red')
279         accf2m=accf2/10
280         aline.set_data(xf, accf2m)
281
282         afitline, = ax.plot([], [],lw=1, color='yellow')
283         fitx=xf[QRstart:fitmax]
284         fity=cosfit(fitx,*popta)
285         fity=sgn*fity/10
286         afitline.set_data(fitx, fity)
287
288
289         leftline, = ax.plot([], [],lw=1, color='yellow')
290         leftline.set_data([xf[wmin],xf[wmin]], [-60,30])
291         rightline, = ax.plot([], [],lw=1, color='yellow')
292         rightline.set_data([xf[wmax],xf[wmax]], [-60,30])
293         plt.show
294
295         # save plot diagrams as png
296         plt.savefig(sdir+"/"+Sess+"-"+str(icnt)+"-"+str(round(T0
297                                     ,2))+".png")
298
299         # write results to file
300         ivdata = open(sdir+"/MaresTableQR.csv",'a');
301
302         ivdata.write(str(Obj)+" ";dt.strftime(dtim+td(seconds=
303                                     int(na[nisp]/fs
304                                     )), '%d.%m.%Y %H
305                                     :%M:%S')+ \
306
307                                     ";"+str(na[nisp])+";"+str(ne[nisp])+ \
308                                     ";"+str(round(T0,2))+";"+str(round(amax,2))+";"+str(
309                                     round(I,3))+ \

```

```

305         ";" + str(round(S, 1)) + ";" + str(round(Sstd, 1)) + ";" + str(
                                                    round(Icosfit
                                                    , 3)) + ";" + str(
                                                    round(Scosfit
                                                    , 1)) + ";" + str(
                                                    round(R2, 4)) +
                                                    ";" + str(
                                                    session) + "\n"
                                                    );
306         ivdata.close()

```

A.3.1.3 *mares_QRanalysis3 – 7_delta_55 – cosfit*

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Tue Mar 24 13:03:13 2020
4
5  @author: mittag
6  """
7  import struct
8  from matplotlib import pyplot as plt
9  import numpy as np
10 from scipy.signal import butter, filtfilt
11 from datetime import datetime as dt
12 from datetime import timedelta as td
13 import os
14 from scipy.optimize import curve_fit
15
16 def lowpass(data, mrange):
17     y = 1 * data
18     for i in range(mrange, len(data) - mrange):
19         y[i] = np.mean(data[i - mrange : i + mrange])
20     return y
21
22 def butter_lowpass_filter(data, cutoff, fs, order):
23     normal_cutoff = cutoff / nyq
24     # Get the filter coefficients
25     b, a = butter(order, normal_cutoff, btype='lowpass', analog=False)
26     y = filtfilt(b, a, data)
27     return y
28
29 def cosfit(x, g, w, c, dx):
30     return g * np.cos(w * (x + dx)) + c
31 def sinfit(x, g, w, c, dx):
32     return g * np.sin(w * (x + dx)) + c
33
34 # prepare result directory

```

```

35 ct = dt.now()
36 ttag=str(ct.year)+'-'+str(ct.month)+'-'+str(ct.day)+'_'+str(ct.hour)+'-'+str
    (ct.minute)
37 sdir='results/'+ttag
38 if not os.path.exists(sdir):
39     os.makedirs(sdir)
40
41 ivdata = open(sdir+"/MaresTableQR.csv",'w');
42 ivdata.write("SubjID;IntervalStartDateTime(UTC);IntervalStartIndex;
    IntervalEndIndex;IntervalMeanTorque
    [Nm];IntervalMaxAngAcceleration[rad
    /sec2];IntervalMomentOfInertia[kgm2
    ];IntervalStiffness[Nm/rad);
    IntvStiffnSTD;MoIfit[kgm2]; Sfit[Nm
    /rad]; File\n")
43 ivdata.close()
44
45 fs=2000
46 nyq=0.5*fs
47 err=-999
48
49 tmlvl=110 # analyse QR intervalls
50
51 #----- generate Mares data table
52 basedirMares="Y:/Pro/AGBRESA1/PI/Rittweger/MARES/dekodiert/Agbresa 1 Ankle 2
    "
53
54 sesss=[]
55 listOfSubjs = os.listdir(basedirMares)
56 #oneSubj=[]
57 #oneSubj.append(listOfSubjs[0])
58 for Subj in listOfSubjs:
59     #for Subj in oneSubj:
60         listOfSesss = os.listdir(basedirMares+"/"+Subj)
61         #oneSess=[]
62         #oneSess.append(listOfSesss[1])
63         for Sess in listOfSesss:
64             #for Sess in oneSess:
65                 dtim=dt.strptime(Sess,'%d%B%Y[%Hh%Mm%Ss]')
66                 session=basedirMares+"/"+Subj+"/"+Sess
67                 print(session)
68                 # read data streams
69
70                 try :
71                     data=open(session+"/Time_Mark_1","rb").read()
72                     trq0=open(session+"/Subject_Torque","rb").read()
73                     pos0=open(session+"/Angular_Position","rb").read()
74                     vel0=open(session+"/Angular_Velocity","rb").read()

```

```

75     except :
76         print("")
77         print("missing data")
78     else :
79         ld=len(data)
80         if ((ld!=len(trq0)) or (ld!=len(pos0)) or (ld!=len(vel0)) ) :
81             print("corrupted data")
82         else :
83
84
85         tm1=[]; trq=[]; pos=[]; vel=[]
86         # convert binary data to double with order big endian
87         for i in range(0,ld,8):
88             rx = struct.unpack('>d',data[i:i+8])[0]; tm1.append(rx) #
89                                     Time_Mark_1
90             rx = struct.unpack('>d',trq0[i:i+8])[0]; trq.append(rx) #
91                                     Subject_Torque
92             rx = struct.unpack('>d',pos0[i:i+8])[0]; pos.append(rx) #
93                                     Angular_Position
94             rx = struct.unpack('>d',vel0[i:i+8])[0]; vel.append(rx) #
95                                     Angular_Velocity
96
97         trq=np.array(trq)
98         trq=butter_lowpass_filter(trq, 20, fs, 4)
99
100        dl=len(tm1) # size of total stream
101
102        na=[]; ne=[]; x=[]
103        x.append(0)
104        for i in range(1,dl): # find QR intervals
105            if (tm1[i] - tm1[i-1]>tmlvl-5 and tm1[i] - tm1[i-1]<tmlvl+5)
106                :
107                na.append(i)
108            if (tm1[i] - tm1[i-1]> -tmlvl-5 and tm1[i] - tm1[i-1]< -
109                tmlvl+5) :
110                ne.append(i)
111            x.append(i) # generation of "time"-stream, every step is 0.5
112                        ms
113
114        lna=len(na) # number of QR intervals found
115        if lna<=0:
116            print("no QR intervals found for this session")
117        else :
118            if (lna-len(ne))==1 :
119                ne.append(na[lna-1]+30000)
120            if ne[lna-1]>len(trq):
121                na.remove(na[lna-1])
122                ne.remove(ne[lna-1])

```

```

116         lna=lna-1
117
118     icnt=0
119     for nisp in range(0,lna): # analyse all intervals
120         icnt=icnt+1
121         cnt=1
122         velspan=-min(vel[na[nisp]:ne[nisp]]) # delta to QR (
                                                # negative)
                                                # maximum
123
124         for i in range(na[nisp]+2, ne[nisp]) :
125             cnt=cnt+1
126             if vel[i]<vel[na[nisp]]-velspan/2 :
127                 qrn timer=cnt # where down-jump is half-way
128                 break
129
130         owin=-300 # offset of plotwindow with respect to qrn timer
131         o0=-40 # offset of estimated 0-point wrt qrn timer
132         ofita=-60 # offset of fit window start wrt qrn timer
133         #ofite=+38 # offset of fit window end wrt qrn timer
134         ofite=+110 # offset of fit window end wrt qrn timer
135         otmeana=-200 #offset of mean torque calc start wrt qrn timer
136         otmeane=-100 #offset of mean torque calc end wrt qrn timer
137
138         # offsets of analysis window around QR event
139         qrn timer=qrn timer+na[nisp]+owin
140         qrn timer=qrn timer+na[nisp]+150
141
142         # estimated offset of 0-point for 1st fit
143         qr0est=qrn timer+na[nisp]+o0
144
145         # offsets of fit windows relative to analysis window
146         ifs=qrn timer+na[nisp]+ofita-qrn timer
147         ife=qrn timer+na[nisp]+ofite-qrn timer
148
149         # define vectors around QR event (for plots)
150         xsp= np.arange(qrn timer, qrn timer)
151         tm1sp=tm1[qrn timer: qrn timer]
152         trqsp=trq[qrn timer: qrn timer]
153         possp=pos[qrn timer: qrn timer]
154         possp=np.array(possps)
155         velsp=vel[qrn timer: qrn timer]
156         velsp=np.array(velsp)
157         accsp2=1*velsp
158         possprad=1*possps/180*np.pi
159         velsp rad=1*velsp/180*np.pi
160
161         # vectors for fit plots

```

```

162         xf=xsp[ifs:ife]-qr0est
163
164         if possprad[3]<0 :
165             possprad=possprad+360;
166         posf=possprad[ifs:ife]
167         velf=velsprad[ifs:ife]
168
169
170
171         # differentiation of pos(x)
172         for i in range(1, len(possprad)):
173             velsp2[i]=(possprad[i]-possprad[i-1])*2000
174         velsp2[0]=0
175         velsp2unf=1*velsp2
176         #velsp2=butter_lowpass_filter(velsp2, 10, fs, 4)
177         velsp2=lowpass(velsp2,3)
178         velsp2=lowpass(velsp2,3)
179         velf2=velsp2[ifs:ife]
180         for i in range(0, len(velf2)):
181             if abs(velf2[i])>0.2:
182                 QRstart=i-4
183                 break
184         if QRstart<0:
185             QRstart=0
186         velf2unf=velsp2unf[ifs:ife]
187         # differentiation of pos(x)
188         for i in range(2, len(possprad)-1):
189             accsp2[i]=(velsp2[i]-velsp2[i-1])*2000
190         accsp2[0]=0
191         accsp2[1]=0
192         accsp2unf=1*accsp2
193         #accsp2=butter_lowpass_filter(accsp2, 30, fs, 3)
194         accsp2=lowpass(accsp2,5)
195         accsp2=lowpass(accsp2,5)
196         accf2=accsp2[ifs:ife]
197         accf2unf=accsp2unf[ifs:ife]
198         amax=min(accf2)
199         amaxpos=np.argmaxmin(accf2)
200         T0=np.mean(trqsp[ifs+otmeana:ifs+otmeane])
201
202         wmin=amaxpos+2
203         #         if wmin>45 :
204         #             wmin=45
205         if wmin>55 :
206             wmin=55
207         wmax= wmin+15
208         #Q=(accf2[amaxpos+25]-accf2[amaxpos+2])/(posf[amaxpos+25]

```

```

209         Q0=[]
210         for i in range(wmin,wmin+21,3):
211             q=(accf2[i+15]-accf2[i])/(posf[i+15]-posf[i])
212             Q0.append(q)
213         #Q=(accf2[wmax]-accf2[wmin])/(posf[wmax]-posf[wmin])
214         Q0=np.array(Q0)
215         Q=np.mean(Q0)
216         Qstd=np.std(Q0)
217
218         I=T0/(amax)
219         S=-Q*I
220         Sstd=Qstd*I
221         #print(amax,amaxpos,S,T0, I)
222
223         #2nd fit of pos(x)
224         fitmin=wmin;
225         fitmax=wmin+36
226         fitxf=xf[fitmin:fitmax]
227         fita=accf2[fitmin:fitmax]
228         ps0=possp[5]
229         sgn=1
230         if amax<0:
231             sgn=-1
232             fita=fita*sgn
233         popta, pcova = curve_fit(cosfit, fitxf, fita, bounds=(0
234                                     .6*sgn*amax,0,-
235                                     0.001,-QRstart+
236                                     19], [2*sgn*
237                                     amax, 0.1, 0.
238                                     001, -QRstart+
239                                     20]))
240
241         print(sgn*amax,popta)
242
243         fitf=cosfit(xf,*popta)
244
245         Qcosfit=(fitf[wmin+15]-fitf[wmin])/(posf[wmin+15]-posf[
246                                     wmin])
247
248         Icosfit=-T0/popta[0]
249         Scosfit=Qcosfit*Icosfit
250
251         # lupe=1
252         # if lupe==1 :
253         #     velspan=velsp2[0]-min(velsp2)
254         #     run=0
255         #     for iq in range(0,qrnove-qrnowa): # analyze all
256         #         intervals
257         #             run=run+1

```



```

248 #             if velsp2[iq]<velsp2[0]-velspan/2 :
249 #                 xmin=qrnowa+run-100
250 #                 xmax=xmin+400
251 #                 break
252 #             else :
253 xmin= min(xf)
254 xmax= max(xf)
255 fig = plt.figure()
256 ax = plt.axes(xlim=(xmin,xmax), ylim=(-60,30))
257 ax.set_xlabel("time [0.5ms]")
258 #ax.set_ylabel('TM1')
259 psline, = ax.plot([], [],lw=1, color='cyan')
260 psline.set_data(xf, (posf-possprad[3])*200)
261 vline, = ax.plot([], [],lw=1, color='blue')
262 velfm=(velf-velsprad[4])*5
263 vline.set_data(xf, velfm)
264
265 v2uline, = ax.plot([], [],lw=1, color='gray')
266 velf2um=(velf2unf-velsp2unf[4])*5
267 v2uline.set_data(xf, velf2um)
268
269 v2line, = ax.plot([], [],lw=1, color='black')
270 velf2m=(velf2-velsp2[4])*5
271 v2line.set_data(xf, velf2m)
272
273 auline, = ax.plot([], [],lw=1, color='green')
274 accf2um=accf2unf/10
275 auline.set_data(xf, accf2um)
276
277 aline, = ax.plot([], [],lw=1, color='red')
278 accf2m=accf2/10
279 aline.set_data(xf, accf2m)
280
281 afitline, = ax.plot([], [],lw=1, color='yellow')
282 fitx=xf[QRstart:fitmax]
283 fity=cosfit(fitx,*popta)
284 fity=sgn*fity/10
285 afitline.set_data(fitx, fity)
286
287
288 leftline, = ax.plot([], [],lw=1, color='yellow')
289 leftline.set_data([xf[wmin],xf[wmin]], [-60,30])
290 rightline, = ax.plot([], [],lw=1, color='yellow')
291 rightline.set_data([xf[wmax],xf[wmax]], [-60,30])
292 plt.show
293
294 plt.savefig(sdir+"/"+Sess+"-"+str(icnt)+"-"+str(round(TO
,2))+".png")

```

```

295
296
297         ivdata = open(sdir+"/MaresTableQR.csv",'a');
298         #C=3
299         ivdata.write(str(Subj)+";"+dt.strftime(dtim+td(seconds=
                                                    int(na[nisp]/fs
                                                    )), '%d.%m.%Y %H
                                                    :%M:%S')+ \
300         ";"+str(na[nisp])+";"+str(ne[nisp])+ \
301         ";"+str(round(T0,2))+";"+str(round(amax,2))+";"+str(
                                                    round(I,3))+
                                                    \
302         ";"+str(round(S,1))+";"+str(round(Sstd,1))+";"+str(
                                                    round(Icosfit
                                                    ,3))+";"+str(
                                                    round(Scosfit
                                                    ,1))+";"+str(
                                                    session)+"\n"
                                                    );
303
303         ivdata.close()

```

A.3.2 SP

A.3.2.1 mares_SP3.py

```

1  # -*- coding: utf-8 -*-
2  """
3  First created on Tue Mar 24 13:03:13 2020
4  version of 17.12.2020
5
6  @author: mittag
7  """
8  import struct
9  from matplotlib import pyplot as plt
10 import numpy as np
11 from scipy.signal import butter, filtfilt
12 from datetime import datetime as dt
13 from datetime import timedelta as td
14 import os
15
16 def butter_lowpass_filter(data, cutoff, fs, order):
17     normal_cutoff = cutoff / nyq
18     # Get the filter coefficients
19     b, a = butter(order, normal_cutoff, btype='low', analog=False)
20     y = filtfilt(b, a, data)
21     return y
22

```

```

23 # recording rate of input data
24 fs=2000
25 nyq=0.5*fs
26 err=-999
27
28 # prepare output folder; use time tag for naming
29 ct = dt.now()
30 ttag=str(ct.year)+'-'+str(ct.month)+'-'+str(ct.day)+'_'+str(ct.hour)+'-'+str
                                     (ct.minute)
31 sdir='results/'+ttag
32 if not os.path.exists(sdir):
33     os.makedirs(sdir)
34
35
36
37 #----- set base directory
38
39 #basedirMares="I:/python_ws/Nikolas/MARES/dekodiert/Agbresa 1 Ankle 2"
40 #basedirMares="I:/Mares/QR/Nils_BA/mares"
41 #basedirMares="D:/Corona_ohne_i/python_ws/qr-sp/AGBRESA/E"
42 basedirMares="G:/MKW/PUB/fuerUwe/Projekt_Sophia_Daten_MARES"
43
44 #prepare output file; write header
45 ivdata = open(sdir+"/MaresSPTable.csv",'w');
46 ivdata.write("#IV;SubjID;IntervalStartDateTime(UTC);IntervalStartIndex;
                                     IntervalEndIndex;IntervalMeanTorque
                                     ;IntervalMeanTorqueAmplitude;
                                     IntervalMeanPositionAmplitude;
                                     IntervalFrequency;
                                     IntervalPhaseShift;
                                     IntervalPhaseShiftcorr;File\n")
47 ivdata.close()
48
49 #----- parse data in base directory
50 sesss=[]
51 listOfSubjs = os.listdir(basedirMares)
52 for Subj in listOfSubjs:
53     listOfSesss = os.listdir(basedirMares+"/"+Subj)
54     for Sess in listOfSesss:
55         dtim=dt.strptime(Sess,'%d%B%Y[%Hh%Mm%Ss]')
56         session=basedirMares+"/"+Subj+"/"+Sess
57         data=open(session+"/Time_Mark_1","rb").read()
58         pos0=open(session+"/Angular_Position","rb").read()
59         trq0=open(session+"/Subject_Torque","rb").read()
60         # convert raw data
61         ld=len(data)
62         tm1=[]
63         trq=[]

```

```

64     pos=[]
65     for i in range(0,ld,8):
66         rx = struct.unpack('>d',data[i:i+8])[0]; tm1.append(rx)
67         rx = struct.unpack('>d',pos0[i:i+8])[0]; pos.append(rx)
68         rx = struct.unpack('>d',trq0[i:i+8])[0]; trq.append(rx)
69     # low pass filter torque input stream
70     trq=np.array(trq)
71     trq=butter_lowpass_filter(trq, 20, fs, 4)
72
73     # low pass filter angular position stream
74     pos=np.array(pos)
75     pos=butter_lowpass_filter(pos, 20, fs, 4)
76
77     dl=len(tm1)
78
79     # identify SP intervals where time_mark_1=70
80     na=[]
81     ne=[]
82     x=[]
83     x.append(0)
84     for i in range(1,dl):
85         if ((tm1[i] - tm1[i-1])>65 and (tm1[i] - tm1[i-1])<75) :
86             na.append(i)
87         if ((tm1[i] - tm1[i-1]> -75) and (tm1[i] - tm1[i-1]< -65)) :
88             ne.append(i)
89             x.append(i)
90
91     if len(na)>0:
92
93         # generate vectors for SP interval
94         xsp= np.arange(na[0], ne[len(ne)-1])
95         tm1sp=tm1[na[0]:ne[len(ne)-1]]
96         possp=pos[na[0]:ne[len(ne)-1]]
97         trqsp=trq[na[0]:ne[len(ne)-1]]
98
99         trqmean=[]
100        trqampl=[]
101        posampl=[]
102        frq=[]
103        pshift=[]
104        pshift2=[]
105        liv=len(na);
106        if liv>len(ne): # number of interval starts differ from number
                        # of interval ends
107
108            liv=len(ne);
109            for i in range(0, liv) : # for all identified intervals
110
                cnttmax=0

```

```

111         cnttmin=0
112         cntpmax=0
113         cntpmin=0
114         tmax=[]
115         tmin=[]
116         tmaxtime=[]
117         pmax=[]
118         pmin=[]
119         pmaxtime=[]
120         rl=int(na[i]+(ne[i]-na[i])*0.6) # set left boundary of
                                         analysis window;
                                         oscillation
                                         sometime start late
                                         therefore use
                                         final 40% of
                                         interval only
121         rr=ne[i]-100 # set right boundary of analysis window; 50ms
                                         prior to interval
                                         end

122         trqv=trq[rl:rr]
123
124         # find torque and position maxima and minima
125         for j in range(rl,rr):
126             if (trq[j-1]<trq[j] and trq[j+1]<trq[j]) :
127                 tmax.append(trq[j]); cnttmax=cnttmax+1
128                 tmaxtime.append(j)
129             if (trq[j-1]>trq[j] and trq[j+1]>trq[j]) :
130                 tmin.append(trq[j]); cnttmin=cnttmin+1
131             if (pos[j-1]<pos[j] and pos[j+1]<pos[j]) :
132                 pmax.append(pos[j]); cntpmax=cntpmax+1
133                 #pmaxtime.append(j)
134             if (pos[j-1]>pos[j] and pos[j+1]>pos[j]) :
135                 pmin.append(pos[j]); cntpmin=cntpmin+1
136                 pmaxtime.append(j)
137
138         mincnt=min(cnttmax,cnttmin,cntpmax,cntpmin) # lowest number
                                                         of maxima/minima
                                                         out of the four
                                                         above
139         dcnt=max(cnttmax,cnttmin,cntpmax,cntpmin)-mincnt # max
                                                         deviation in number
                                                         between maxima

140         frqi=mincnt/(rr-rl)*fs
141
142         tmax=np.array(tmax)
143         tmin=np.array(tmin)
144         pmax=np.array(pmax)
145         pmin=np.array(pmin)

```

```

146         if (frqi < 20 or frqi > 1) and dcnt<6: # check consistency;
                                                    if frequency out of
                                                    range of number of
                                                    maxima differ
                                                    strongly ignore
                                                    interval
147         # calculate amplitudes; use first mincnt oscillations
                                                    only
148         tmaxu=tmax[0:mincnt]
149         tminu=tmin[0:mincnt]
150         tampl=tmaxu-tminu
151         trqampl.append(np.mean(tampl)) # vector of mean torque
                                                    amplitudes (
                                                    deltas of
                                                    neighbored
                                                    maxima/minima)

152         pmaxu=pmax[0:mincnt]
153         pminu=pmin[0:mincnt]
154         pampl=pmaxu-pminu
155         posampl.append(np.mean(pampl)) # vector of mean angular
                                                    position
                                                    amplitudes (
                                                    deltas of
                                                    neighbored
                                                    maxima/minima)

156
157         pshiftu=(np.array(pmaxtime[0:mincnt])-np.array(tmaxtime[
                                                    0:mincnt])) #
                                                    array of mincnt
                                                    shifts between
                                                    position and
                                                    torque phase
158         pshiftm=np.mean(pshiftu)*frqi/fs*360 # mean shift over
                                                    mincnt
                                                    oscillations

159         pshiftm2=1*pshiftm
160         if pshiftm2>100: # if shift is high positive, correct by
                                                    360
161             pshiftm2=pshiftm2-360
162         pshift.append(pshiftm) # vector of phase shift
163         pshift2.append(pshiftm2) # vector of corrected phase-
                                                    shift
164         trqmean.append(np.mean(trqv)) # vector of mean torque
165         frq.append(frqi)
166     else : # in case of bad quality set values to -999
167         trqmean.append(np.mean(trqv))
168         trqampl.append(err)
169         posampl.append(err)

```

```

170         frq.append(err)
171         pshift.append(err)
172         pshift2.append(err)
173
174     # write data to file
175     ivdata = open(sdir+"/MaresSPTable.csv",'a');
176     for i in range(0, liv) :
177         ivdata.write(str(i)+";" + str(Subj)+";" + dt.strftime(dtim+td(
                                                    seconds=int(na[i]/
                                                    fs)), '%d.%m.%Y %H:%
                                                    M:%S')+ \
178
179         ";" + str(na[i]) + ";" + str(ne[i]) + \
180         ";" + str(round(trqmean[i],2)) + ";" + str(round(trqampl[i],2)) +
                                                    ";" + str(round(
                                                    posampl[i],2)) + \
181         ";" + str(round(frq[i],2)) + ";" + str(round(pshift[i],1)) + ";" +
                                                    str(round(pshift2
                                                    [i],1)) + ";" + str(
                                                    session)+"\n");
181     ivdata.close()

```

A.3.2.2 bode.py

```

1  from datetime import datetime
2  from scipy import signal
3  import math
4  import control as ctl
5  import numpy as np
6  import matplotlib.pyplot as plt
7  import csv as csvlib
8  import os
9
10 # MACT-Toleranz [ ]
11 MACTtol = 0.2
12 # Frequenztoleranz [Hz]
13 Ftol = 0.3
14 # MVC-Toleranz [%]
15 MVCtol = 0.1
16 # Massentraegheitsmoment [kgm^2]
17 I = 0.223
18 # Daempfungskonstante Drehbewegung [Nms]
19 B = [0, 105.0]
20 # Federkonstante (low, mid, high) [rad/Nm]
21 K = [158.14, 248.43, 478.08]
22 # MVC [Nm]
23 MVC = [80, 120]
24 # Muskelaktivierung (0, 0.25, 0.5, 0.75) [ ]

```

```

25 MACT = [0, 0.25, 0.5, 0.75]
26 # Frequenzen [Hz]
27 F = [4, 5, 6, 7, 8, 9, 10, 12, 14, 16]
28 # Typen
29 TYPE_PYTHON = "Python"
30 TYPE_AGBRESA = "AGBRESA"
31 TYPE_PHANTOM = "Phantom"
32 TYPE = [TYPE_AGBRESA, TYPE_PHANTOM]
33
34 SHOWMEAN_MEAN = "mean"
35 SHOWMEAN_NOMEAN = "nomean"
36 SHOWMEAN_BOTH = "both"
37
38 COLORS = ["#FF0000", "#56a03c", "#0000FF"] # Colors used in plotting
39
40 MAGNPHASE_MAGN = "magn"
41 MAGNPHASE_PHASE = "phase"
42 MAGNPHASE = [MAGNPHASE_MAGN, MAGNPHASE_PHASE]
43
44 FOCUS_MVC = "MVC"
45 FOCUS_PERCENTMVC = "%MVC"
46 FOCUS_K = "K"
47 FOCUS_B = "B"
48 FOCUS_BANDK = "B UND K"
49
50 # Initialisierung und Labeling des Arrays
51 offset = 3 * len(B) * len(K) * len(MACT) * len(MVC)
52 rows, cols = (2 * offset + 1, len(F) + 6)
53 csv = [[0 for i in range(cols)] for j in range(rows)]
54 csv[0][0] = "Type"
55 csv[0][1] = "Magn/Phase"
56 csv[0][2] = "MACT"
57 csv[0][3] = "K"
58 csv[0][4] = "B"
59 csv[0][5] = "MVC"
60 index_col = 6
61 index_row = 0
62 for f in F:
63     csv[0][index_col] = str(f)
64     index_col += 1
65
66 Fmath = []
67 for j in range(400, 1600, 1):
68     Fmath.append(j/100)
69 print(Fmath)
70
71 def roundValue(val, tol):

```



```

72     """ Round given value if within tolerance. Returns -1 if outside
73         tolerance. """
74     preint = math.floor(val)
75     postint = math.floor(val + tol)
76     if preint != postint:
77         return postint
78     postint = math.floor(val - tol)
79     if preint != postint:
80         return preint
81     return -1
82
83 def isInRange(val, comparisonval, tol):
84     """ Check if value is in range of comparison value and its tolerance """
85     return comparisonval * (1 - tol) <= val <= comparisonval * (1 + tol)
86
87
88 def generateKey(type, Magnphase, B, K=0, MACT=0, MVC=0):
89     """ Generate key for dictionary from combination of Type, Magn/Phase, B,
90         K, MACT and MVC """
91     return "" + str(type) + "X" + str(Magnphase) + "X" + str(float(B)) + "X"
92         + str(float(K)) + "X" + str(
93             float(MACT)) + "X" + str(float(MVC))
94
95
96 def readCsv(cachedict, inputcsvname, type):
97     """ Open and read values of input csv
98
99     make basic sanity checks
100
101     round MACT and decide if MACT is within range of 0/0.25/0.5/0.75
102
103     check if AGBRESA or Phantom data
104     if AGBRESA: get data for specified target MVC (80/120)
105
106     read frequency value
107
108     convert magnitude into dB-scale
109
110     convert phase into Hz-values
111
112     generate array of magnitudes sorted by dictionary
113
114     generate array of phases sorted by dictionary
115     """
116     with open(inputcsvname, newline="\n") as csvfile:
117         spamreader = csvlib.reader(csvfile, delimiter=';', quotechar='|')
118         index_row = 0

```

```

117     lastkeymagn = -1
118     lastkeyphase = -1
119     for row in spamreader:
120         if index_row > 1 and float(row[6]) > -999 and float(row[7]) > 0:
121             # basic sanity checks
122             # MACT lesen
123             if float(row[14]) != 0:
124                 mact = roundValue(4 * (abs(float(row[5]) / float(row[14]
125                                     ))) , MACTtol) /
126                                     4
127
128                 if mact == -1:
129                     continue
130                 else:
131                     continue
132
133             # MVC und b und k lesen
134             if type == TYPE_AGBRESA:
135                 if isInRange(abs(float(row[14])), 80, MVCtol):
136                     mvc = 80
137                 elif isInRange(abs(float(row[14])), 120, MVCtol):
138                     mvc = 120
139                 else:
140                     continue
141                 k = 0
142                 b = 0
143             else:
144                 mvc = float(row[14])
145                 k = float(row[12])
146                 b = float(row[13])
147
148             # Frequenzwert lesen
149             freq = roundValue(float(row[8]), Ftol)
150             if freq == -1:
151                 continue
152             magn = 20 * math.log10(float(row[7])*math.pi/180 / float(row
153                                     [6]))
154
155             phase = float(row[10])
156
157             # Magnitude schreiben
158             key = generateKey(type, "magn", b, k, mact, mvc)
159             if not key in cachedict:
160                 cachedict[key] = []
161             if key != lastkeymagn:
162                 cachedict[key].append([])
163             cachedict[key][len(cachedict[key]) - 1].append([magn, float(
164                                     row[8])])
165
166             lastkeymagn = key

```

```

160         # Phase schreiben
161         key = generateKey(type, "phase", b, k, mact, mvc)
162         if not key in cachedict:
163             cachedict[key] = []
164         if key != lastkeyphase:
165             cachedict[key].append([])
166         cachedict[key][len(cachedict[key]) - 1].append([phase, float
                                                         (row[8])])
167
168         lastkeyphase = key
169
170         index_row += 1
171         #print(cachedict)
172
173 def convertCacheDict(cachedict):
174     """ convert array to dictionary
175
176     choose most complete array
177
178     transfer most complete array to subdictionary"""
179     cachedictout = {}
180     for key in cachedict.keys():
181         longestseq = None
182         previousseqlen = -1
183         for seq in cachedict[key]: # Suchen der laengsten Sequenz
184             if len(seq) >= previousseqlen:
185                 longestseq = seq
186                 previousseqlen = len(seq)
187         cachedictout[key] = {}
188         for sequele in longestseq: # Uebertragen der laengsten Sequenz in
                                     ein SubCacheDict
189             freq = roundValue(sequele[1], Ftol)
190             cachedictout[key][freq] = sequele[0]
191     return cachedictout
192
193
194 def labelRows(index_row, type, magphase, mact, k, b, mvc):
195     """ label output-csv"""
196     csv[index_row][0] = type
197     csv[index_row][1] = magphase
198     csv[index_row][2] = mact
199     csv[index_row][3] = k
200     csv[index_row][4] = b
201     csv[index_row][5] = mvc
202
203
204 def writeFreqToCsv(csv, index_row, cacheDict, key):
205     """ write values to output-csv

```

```

206
207     if no values present or outside tolerance
208         write no data"""
209     index_col = 6
210     for f in F:
211         if key not in cacheDict or f not in cacheDict[key]:
212             csv[index_row][index_col] = "no data"
213         else:
214             csv[index_row][index_col] = cacheDict[key][f]
215         index_col += 1
216
217
218 def buildCsv(csv, cachedict):
219     """ use dictionary to fill output-csv"""
220     index_row = 3
221     # Berechnung aller Permutationen aus B, K und MACT
222     for b in B:
223         for mvc in MVC:
224             for k in K:
225                 for mact in MACT:
226                     # Labeling des Arrays
227                     labelRows(index_row, TYPE_PYTHON, "Magn", mact, k, b,
228                                mvc)
229                     labelRows(index_row - 1, TYPE_AGBRESA, "Magn", mact, k,
230                                b, mvc)
231                     labelRows(index_row - 2, TYPE_PHANTOM, "Magn", mact, k,
232                                b, mvc)
233                     labelRows(index_row + offset, TYPE_PYTHON, "Phase", mact
234                                , k, b, mvc)
235                     labelRows(index_row + offset - 1, TYPE_AGBRESA, "Phase",
236                                mact, k, b,
237                                mvc)
238                     labelRows(index_row + offset - 2, TYPE_PHANTOM, "Phase",
239                                mact, k, b,
240                                mvc)
241
242                     mag, phase = calcMagPhase(k, b, F)
243                     print("calcMagPhase(", k, ",", b, ",", [0]): ",
244                            calcMagPhase(k,
245                            b, [0]))
246
247                     # Eintragen der errechneten Magnitude und Phase in die
248                        output-csv
249
250                     index_col = 6
251                     for m in mag:
252                         csv[index_row][index_col] = m
253                         index_col += 1
254                     index_col = 6

```

```

243         for p in phase:
244             csv[index_row + offset][index_col] = p
245             index_col += 1
246
247         key = generateKey(TYPE_PHANTOM, "magn", b, k, mact, mvc)
248         writeFreqToCsv(csv, index_row - 2, cachedict, key)
249
250         key = generateKey(TYPE_PHANTOM, "phase", b, k, mact, mvc
251                             )
252         writeFreqToCsv(csv, index_row + offset - 2, cachedict,
253                             key)
254
255         key = generateKey(TYPE_AGBRESA, "magn", 0, 0, mact, mvc)
256         writeFreqToCsv(csv, index_row - 1, cachedict, key)
257
258         key = generateKey(TYPE_AGBRESA, "phase", 0, 0, mact, mvc
259                             )
260         writeFreqToCsv(csv, index_row + offset - 1, cachedict,
261                             key)
262
263         index_row += 3
264
265     def standardDeviation(cachedict, type, focus, magnphase):
266         """
267         """
268         perms = []
269         mact_sub = [0.25, 0.5, 0.75]
270         if focus == FOCUS_B: #Korrekte Reihung der Variablen entsprechend des
271                                 Fokus
272             perms = [MVC, K, mact_sub, B]
273         elif focus == FOCUS_K:
274             perms = [MVC, B, mact_sub, K]
275         elif focus == FOCUS_MVC:
276             perms = [B, K, mact_sub, MVC]
277         elif focus == FOCUS_PERCENTMVC:
278             perms = [MVC, K, B, mact_sub]
279         results = []
280         for p0 in perms[0]: # For-Verschachtelung entsprechend der oben
281                                 festgelegten Reihung
282             for p1 in perms[1]:
283                 for p2 in perms[2]:
284                     allfreqs = []
285                     for p3 in perms[3]:
286                         if focus == FOCUS_B: # Zuordnung der Iterationsvariablen
287                             b = p3
288                             k = p1
289                             mact = p2

```

```

285         mvc = p0
286     elif focus == FOCUS_K:
287         b = p1
288         k = p3
289         mact = p2
290         mvc = p0
291     elif focus == FOCUS_MVC:
292         b = p0
293         k = p1
294         mact = p2
295         mvc = p3
296     elif focus == FOCUS_PERCENTMVC:
297         b = p2
298         k = p1
299         mact = p3
300         mvc = p0
301     if type == TYPE_AGBRESA: #Pruefung ob AGBRESA-Daten
                                vorliegen und
                                der Fokus auf B
                                oder K liegt
302         if focus == FOCUS_B or focus == FOCUS_K:
303             #print("Keine b- oder k-Zuordnung moeglich fuer
                                AGBRESA
                                !")
304         return
305         b = 0
306         k = 0
307     key = generateKey(type, magnphase, b, k, mact, mvc) #
                                Erstellung der
                                Keys
                                entsprechend
                                der
                                Iterationsvariablen
308
309     if key in cachedict: #ueberpruefung ob der key im
                                dictionary
                                vorhanden ist
310
311         freqs = []
312         for f in F: # iteration ueber F
313             if f in cachedict[key]: # Ueberpruefung ob ein
                                    Wert
                                    zur
                                    Frequenz
                                    im
                                    cachedict
                                    [key]
                                    vorhanden
                                    ist

```

```

312         freqs.append(cachedict[key][f]) # falls ja
                                           ->
                                           anhaengen
                                           des
                                           Wertes
                                           an
                                           freqs
                                           -
                                           array

313     else:
314         freqs.append(np.nan) #falls nein -> NaN
                                           setzen

315         allfreqs.append(freqs) #anhaengen des freqs-arrays
                                           in das
                                           ueberarray
                                           "allfreqs"

316     stddevs = []
317     for j in range(len(F)): #iteration ueber die laenge von F
318         freqcolumn = []
319         for i in range(len(allfreqs)): #iteration ueber die
                                           laenge von
                                           allfreqs

320             if allfreqs[i][j] is not np.nan: #sofern das i-te
                                           element im
                                           jeweiligen
                                           allfreqs-
                                           array nicht
                                           NaN ist...

321                 freqcolumn.append(allfreqs[i][j]) #... anhaengen
                                           des
                                           wert
                                           an
                                           freqcolumn

322             if len(freqcolumn) > 0: # unter der bedingung, dass
                                           freqcolumn
                                           mindestens
                                           einen Wert hat
                                           ...

323                 stddevs.append(np.std(freqcolumn)) # berechne die
                                           standardabweichung
                                           der werte
                                           der

```

```

324         if len(stddevs) > 0: #sofern stddevs mindestens einen wert
325             results.append(stddevs) #anhaengen des stddevs-arrays (
326                 avgstddevs = []
327                 for j in range(len(F)): #iteration ueber die laenge von F
328                     stddevcolumn = []
329                     for i in range(len(results)):
330                         if j < len(results[i]): #da z.T. nicht so viele werte in results
331                             stddevcolumn.append(results[i][j]) #sammeln der werte jedes
332                             avgstddevs.append(np.average(stddevcolumn))#anhaengen des
333
334
335
336 def calcMagPhase(k, b, xvalues):
337
338     # Coefficients in numerator of transfer function
339     num = [1]
340     # Coefficients in denominator of transfer function
341     # High order to low order, eg 1*s^2 + 0.1*s + 1
342     den = [1, b, k]
343     s1 = signal.lti(num, den)
344     # Specify our own frequency range: np.arange(0.1, 5, 0.01)
345     freqconv = np.array(xvalues) * 2 * math.pi
346     w, mag, phase = signal.bode(s1, freqconv.tolist())
347
348     return mag, phase
349
350
351 def appendToDict(dict, key, value):

```



```
384         continue
385         b = 0
386         k = 0
387         key = generateKey(type, magnphase, b, k, mact, mvc)
                                     #
                                     Erstellung
                                     der Keys
                                     entsprechend
                                     der
                                     Iterationsvariablen

388         if key in cachedict: # ueberpruefung ob der key im
                                     dictionary
                                     vorhanden
                                     ist

389         freqs = []
390         for f in F: # iteration ueber F
391             if f in cachedict[key]: # Ueberpruefung ob
                                     ein
                                     Wert
                                     zur
                                     Frequenz
                                     im
                                     cachedict
                                     [
                                     key
                                     ]
                                     vorhanden
                                     ist

392         freqs.append(cachedict[key][f]) # falls
                                     ja
                                     ->
                                     anhaengen
                                     des
                                     Wertes
                                     an
```

```

393         else:
394             freqs.append(np.nan) # falls nein -> "
                                     nan
                                     "
                                     setzen
395
396         if focus == FOCUS_BANDK:
397             allFreqsKey = generateKey(type, magnphase, b
                                     , k
                                     )
398         else:
399             allFreqsKey = generateKey(type, magnphase,
                                     focusvar
                                     )
400
401             appendToDict(allfreqs, allFreqsKey, freqs)
402
403         mag, phase = calcMagPhase(k, b, Fmath)
404         magconv = []
405         for m in mag:
406             magconv.append(m)
407         if focus == FOCUS_BANDK:
408             allFreqsKey = generateKey(TYPE_PYTHON,
                                     MAGNPHASE_MAGN
                                     , b, k)
409         else:
410             allFreqsKey = generateKey(TYPE_PYTHON,
                                     MAGNPHASE_MAGN
                                     ,
                                     focusvar
                                     )
411
412             appendToDict(allfreqs, allFreqsKey, magconv)
413
414         phaseconv = []
415         for p in phase:
416             phaseconv.append(p)
417         if focus == FOCUS_BANDK:
418             allFreqsKey = generateKey(TYPE_PYTHON,
                                     MAGNPHASE_PHASE
                                     , b, k)
419         else:
420             allFreqsKey = generateKey(TYPE_PYTHON,
                                     MAGNPHASE_PHASE
                                     ,
                                     focusvar
                                     )

```

```

417         )
418         appendToDict(allfreqs, allFreqsKey, phaseconv)
419
420     meanalpha = "11"
421     if (type == TYPE_AGBRESA):
422         meanalpha = "06"
423     if focus == FOCUS_BANDK:
424         index_x = 0
425         for k in K:
426             index_y = 0
427             for magnphase in MAGNPHASE:
428                 allFreqsKey = generateKey(type, magnphase, focusb, k)
429
430                 if showmean == SHOWMEAN_MEAN:
431                     plotFreqs(ax[index_y][index_x], allfreqs, allFreqsKey, F
432                             , "r", True,
433                             True, type)
434
435                 elif showmean == SHOWMEAN_NOMEAN:
436                     plotFreqs(ax[index_y][index_x], allfreqs, allFreqsKey, F
437                             , "r", True,
438                             False, type)
439
440                 elif showmean == SHOWMEAN_BOTH:
441                     plotFreqs(ax[index_y][index_x], allfreqs, allFreqsKey, F
442                             , "r", True,
443                             True, type)
444
445                 plotFreqs(ax[index_y][index_x], allfreqs, allFreqsKey, F
446                             , "#FF0000" +
447                             meanalpha, True
448                             , False, type)
449
450                 allFreqsKey = generateKey(TYPE_PYTHON, magnphase, focusb, k)
451                 plotFreqs(ax[index_y][index_x], allfreqs, allFreqsKey, Fmath
452                             , "black", False,
453                             False, "math.
454                             Modell")
455
456             index_y += 1
457
458         ax[0, index_x].set_title(str(k) + " Nm")
459         ax[1, index_x].set_xlabel("Frequenz [Hz]")
460         ax[0, 2].legend()
461         index_x += 1
462     ax[0, 0].set_ylabel("Gain [dB]")
463     ax[1, 0].set_ylabel("Phase [deg]")
464
465     fig.suptitle(type + " - B = " + str(focusb) + " Nms/rad")
466 else:
467     index_p3y = 0

```

```

452     for magnphase in MAGNPHASE:
453         index_color = 0
454         if focus == FOCUS_PERCENTMVC:
455             focusarr = mact_sub
456         elif focus == FOCUS_B:
457             focusarr = B
458         elif focus == FOCUS_K:
459             focusarr = K
460         elif focus == FOCUS_MVC:
461             focusarr = MVC
462         for focusvar in focusarr:
463             allFreqsKey = generateKey(type, magnphase, focusvar)
464             linecolor = COLORS[index_color]
465             if showmean == SHOWMEAN_MEAN:
466                 plotFreqs(ax[index_p3y], allfreqs, allFreqsKey, F,
                                     linecolor, True
                                     , True, str(
                                     focusvar)+
                                     unitlabel)
467
468             elif showmean == SHOWMEAN_NOMEAN:
469                 plotFreqs(ax[index_p3y], allfreqs, allFreqsKey, F,
                                     linecolor, True
                                     , False, str(
                                     focusvar)+
                                     unitlabel)
469
470             elif showmean == SHOWMEAN_BOTH:
471                 plotFreqs(ax[index_p3y], allfreqs, allFreqsKey, F,
                                     linecolor, True
                                     , True, str(
                                     focusvar)+
                                     unitlabel)
471
472                 plotFreqs(ax[index_p3y], allfreqs, allFreqsKey, F,
                                     linecolor +
                                     meanalpha, True
                                     , False, str(
                                     focusvar)+
                                     unitlabel)
472
473         index_color += 1
474
475         allFreqsKey = generateKey(TYPE_PYTHON, magnphase, focusvar)
476         #plotFreqs(ax[index_p3y], allfreqs, allFreqsKey, Fmath, "
477                                     black", False,
478                                     False, focusvar)
476
477
478         index_p3y += 1
479     ax[0].legend()
480     ax[0].set_title(focus)

```

```

481         ax[0].set_ylabel("Gain [dB]")
482         ax[1].set_ylabel("Phase [deg]")
483         ax[1].set_xlabel("Frequenz [Hz]")
484         fig.suptitle(type)
485         fig.savefig(newpath + "/" + type + "_" + focus + "_" + str(focusb) + ".
                                     png")
486
487
488     def plotFreqs(ax, allfreqs, allFreqsKey, xvalues, linecolor, clean, mean,
                                     label=""):
489         if allFreqsKey in allfreqs:
490             showlegend = True
491             if mean:
492                 addarray = [0] * len(allfreqs[allFreqsKey][0])
493                 counter = [0] * len(allfreqs[allFreqsKey][0])
494                 for arr in allfreqs[allFreqsKey]:
495                     index = 0
496                     if clean:
497                         arr = cleanuparray(arr, -250)
498                     for ele in arr:
499                         if ele is not np.nan:
500                             addarray[index] += ele
501                             counter[index] += 1
502                     index += 1
503                 meanarray = []
504                 for i in range(len(addarray)):
505                     if counter[i] != 0:
506                         meanarray.append(addarray[i] / counter[i])
507                     else:
508                         meanarray.append(np.nan)
509                 ax.plot(xvalues, meanarray, color=linecolor, label="meansymbol"+
                                     str(label))
510                 #print("Meanarray fuer " + allFreqsKey + " ist gleich: ",
                                     meanarray)
511
512             else:
513                 for arr in allfreqs[allFreqsKey]:
514                     label = label if showlegend else ""
515                     showlegend = False
516                     if clean:
517                         ax.plot(xvalues, cleanuparray(arr, -250), color=
                                     linecolor,
                                     label=label)
518                     else:
519                         ax.plot(xvalues, arr, color=linecolor, label=label)
520                 ax.grid(True)
521             else:
522                 print("allfreqs fuer key: ", allFreqsKey, " nicht vorhanden")

```

```

523
524     #plt.legend() # Einfuegen der Legende in die erstellten Plots
525
526
527 def cleanuparray(array, cleanupcutoff):
528     results = []
529     for ele in array:
530         if ele < cleanupcutoff:
531             results.append(np.nan)
532         else:
533             results.append(ele)
534     return results
535
536
537 def writeCsvToFile(csv, fileName):
538     """ transfer csv to actual file"""
539     # Uebertragen des Arrays in eine .csv-Datei
540     file1 = open(fileName, "w")
541     for i in range(rows):
542         for j in range(cols):
543             file1.write(str(csv[i][j]))
544             if j < cols - 1:
545                 file1.write("; ")
546             file1.write("\n")
547     file1.close()
548
549
550 cachedict = {}
551 readCsv(cachedict, "MaresSPTable_high_105.0.csv", TYPE_PHANTOM) # read
552 input csv
553 readCsv(cachedict, "MaresSPTable2.11113.csv", TYPE_AGBRESA)
554 cachedictout = convertCacheDict(cachedict) # pick best sequences, filter
555 bad ones
556 buildCsv(csv, cachedictout) # build output csv array
557
558 newpath = str(datetime.now().strftime('%Y-%m-%dT%H_%M')) # create output
559 folder
560
561 if not os.path.exists(newpath):
562     os.makedirs(newpath)
563 writeCsvToFile(csv, newpath + "/bode_plot" + ".csv") # create output csv
564
565 plotting(cachedictout, FOCUS_PERCENTMVC, TYPE_PHANTOM, SHOWMEAN_BOTH)
566 plotting(cachedictout, FOCUS_B, TYPE_PHANTOM, SHOWMEAN_BOTH)
567 plotting(cachedictout, FOCUS_MVC, TYPE_PHANTOM, SHOWMEAN_BOTH)
568 plotting(cachedictout, FOCUS_K, TYPE_PHANTOM, SHOWMEAN_BOTH)
569 plotting(cachedictout, FOCUS_PERCENTMVC, TYPE_AGBRESA, SHOWMEAN_BOTH)
570 plotting(cachedictout, FOCUS_MVC, TYPE_AGBRESA, SHOWMEAN_BOTH)
571 for b in B:

```

```
568     plotting(cachedictout, FOCUS_BANDK, TYPE_PHANTOM, SHOWMEAN_BOTH, b)
569
570 #print("stddevs magn phantom mact: ", standardDeviation(cachedictout,
571                                                         TYPE_PHANTOM, FOCUS_PERCENTMVC, "
572                                                         magn"))
571 #print("stddevs magn phantom b: ", standardDeviation(cachedictout,
572                                                         TYPE_PHANTOM, FOCUS_B, "magn"))
572 #print("stddevs magn phantom k: ", standardDeviation(cachedictout,
573                                                         TYPE_PHANTOM, FOCUS_K, "magn"))
573 #print("stddevs magn phantom mvc: ", standardDeviation(cachedictout,
574                                                         TYPE_PHANTOM, FOCUS_MVC, "magn"))
574 #print("stddevs magn agbresa mact: ", standardDeviation(cachedictout,
575                                                         TYPE_AGBRESA, FOCUS_PERCENTMVC, "
576                                                         magn"))
575 #print("stddevs magn agbresa mvc: ", standardDeviation(cachedictout,
576                                                         TYPE_AGBRESA, FOCUS_MVC, "magn"))
576 #print("stddevs phase phantom mact: ", standardDeviation(cachedictout,
577                                                         TYPE_PHANTOM, FOCUS_PERCENTMVC, "
578                                                         phase"))
577 #print("stddevs phase phantom b: ", standardDeviation(cachedictout,
579                                                         TYPE_PHANTOM, FOCUS_B, "phase"))
578 #print("stddevs phase phantom k: ", standardDeviation(cachedictout,
579                                                         TYPE_PHANTOM, FOCUS_K, "phase"))
579 #print("stddevs phase phantom mvc: ", standardDeviation(cachedictout,
580                                                         TYPE_PHANTOM, FOCUS_MVC, "phase"))
580 #print("stddevs phase agbresa mact: ", standardDeviation(cachedictout,
581                                                         TYPE_AGBRESA, FOCUS_PERCENTMVC, "
582                                                         phase"))
581 #print("stddevs phase agbresa mvc: ", standardDeviation(cachedictout,
582                                                         TYPE_AGBRESA, FOCUS_MVC, "phase"))
582 plt.show()    # Diagramm inklusive Legende
```